Proceedings of Machine Learning Research vol 195:1–35, 2023 36th Annual Conference on Learning Theory

Accelerated Riemannian Optimization: Handling Constraints with a Prox to Bound Geometric Penalties

David Martínez-Rubio MARTINEZ-RUBIO@ZIB.DE
Zuse Institute Berlin and Technische Universität
Berlin, Germany

Sebastian Pokutta POKUTTA@ZIB.DE
Zuse Institute Berlin and Technische Universität
Berlin, Germany

Editors: Gergely Neu and Lorenzo Rosasco

Abstract

We propose a globally-accelerated, first-order method for the optimization of smooth and (strongly or not) geodesically-convex functions in a wide class of Hadamard manifolds. We achieve the same convergence rates as Nesterov's accelerated gradient descent, up to a multiplicative geometric penalty and log factors. Crucially, we can enforce our method to stay within a compact set we define. Prior fully accelerated works *resort to assuming* that the iterates of their algorithms stay in some pre-specified compact set, except for two previous methods of limited applicability. For our manifolds, this solves the open question in (Kim and Yang, 2022) about obtaining global general acceleration without iterates assumptively staying in the feasible set.

In our solution, we design an accelerated Riemannian inexact proximal point algorithm, which is a result that was unknown even with exact access to the proximal operator, and is of independent interest. For smooth functions, we show we can implement the prox step inexactly with first-order methods in Riemannian balls of certain diameter that is enough for global accelerated optimization.

1. Introduction

Riemannian optimization concerns the optimization of a function defined over a Riemannian manifold. It is motivated by constrained problems that can be naturally expressed on Riemannian manifolds allowing to exploit the geometric structure of the problem and effectively transforming it into an unconstrained one. Moreover, there are problems that are not convex in the Euclidean setting, but that when posed as problems over a manifold with the right metric, are convex when restricted to every geodesic, and this allows for fast optimization (Cruz Neto et al., 2006; Carvalho Bento and Melo, 2012; Bento et al., 2015; Allen-Zhu et al., 2018). That is, they are geodesically convex (g-convex) problems, cf. Definition 1. Some applications of Riemannian optimization in machine learning robust covariance estimation in Gaussian distributions (Wiesel, 2012), Gaussian mixture models (Hosseini and Sra, 2015), operator scaling (Allen-Zhu et al., 2018), computation of Brascamp-Lieb constants (Bennett et al., 2008), Karcher mean (Zhang et al., 2016), Wasserstein Barycenters (Weber and Sra, 2017), include dictionary learning (Cherian and Sra, 2017; Sun et al., 2017), low-rank matrix completion (Vandereycken, 2013; Mishra and Sepulchre, 2014; Tan et al., 2014; Cambier

. Most of the notations in this work have a link to their definitions. For example, if you click or tap on any instance of x^*, you will jump to the place where it is defined as the global minimizer of the function we consider in this work.

and Absil, 2016; Heidel and Schulz, 2018), optimization under orthogonality constraints (Edelman et al., 1998; Lezcano-Casado and Martínez-Rubio, 2019), and sparse principal component analysis (Jolliffe et al., 2003; Genicot et al., 2015; Huang and Wei, 2019b). The first seven problems are defined over Hadamard manifolds, which we consider in this work, and the first six are g-convex problems to which our results can be applied. In fact, the optimization in these cases is over symmetric spaces, which satisfy a property that one instance of our algorithm requires, cf. Theorem 6.

Riemannian optimization, whether under g-convexity or not, is an extensive and active area of research, for which one aspires to develop Riemannian optimization algorithms that share analogous properties to the more broadly studied Euclidean methods, such as the following kinds of Riemannian first-order methods: deterministic (Wei et al., 2016; Zhang and Sra, 2016; Bento et al., 2017), adaptive (Kasai et al., 2019), projection-free (Weber and Sra, 2017, 2019), saddle-point-escaping (Criscitiello and Boumal, 2019; Sun et al., 2019; Zhou et al., 2019; Criscitiello and Boumal, 2020), stochastic (Hosseini and Sra, 2017; Khuzani and Li, 2017; Tripuraneni et al., 2018), variance-reduced (Zhang et al., 2016; Sato et al., 2017, 2019), and min-max methods (Zhang et al., 2022; Jordan et al., 2022), among others.

Riemannian generalizations to accelerated convex optimization are appealing due to their better convergence rates with respect to unaccelerated methods, specially in ill-conditioned problems. Acceleration in Euclidean convex optimization is a concept that has been broadly explored and has provided many different fast algorithms. A paradigmatic example is Nesterov's Accelerated Gradient Descent (AGD), cf. (Nesterov, 1983), which is considered the first general accelerated method, where the conjugate gradients method can be seen as an accelerated predecessor in a more limited scope (Martínez-Rubio, 2020). There have been recent efforts to better understand this phenomenon in the Euclidean case (Drori and Teboulle, 2014; Su et al., 2016; Wibisono et al., 2016; Allen Zhu and Orecchia, 2017; Diakonikolas and Orecchia, 2019; Joulani et al., 2020), which have yielded some fruitful techniques for the general development of methods and analyses. These techniques have allowed for a considerable number of new results going beyond the standard oracle model, convexity, or beyond first-order, in a wide variety of settings (Tseng, 2008; Beck and Teboulle, 2009; Wang et al., 2016a; Allen Zhu and Orecchia, 2015; Allen-Zhu, 2017, 2018; Carmon et al., 2017; Diakonikolas and Orecchia, 2018; Hinder et al., 2019; Gasnikov et al., 2019; Kamzolov and Gasnikov, 2020; Ivanova et al., 2021; Criado et al., 2021), among many others. There have been some efforts to achieve acceleration for Riemannian algorithms as generalizations of AGD, cf. Section 3. These works try to answer the following fundamental question:

Can a Riemannian first-order method enjoy the same rates of convergence as Euclidean AGD?

The question is posed under (possibly strongly) geodesic convexity and smoothness of the function to be optimized. And due to the lower bound in (Criscitiello and Boumal, 2021), we know the optimization must be under bounded geodesic curvature of the Riemannian manifold, and we might have to optimize over a bounded domain.

Main results In this work, we study the question above in the case of finite-dimensional Hadamard manifolds $\mathcal{M}$ of bounded sectional curvature and provide an instance of our framework for a wide class of Hadamard manifolds. For a function $f : \mathcal{M} \to \mathbb{R}$ with a global minimizer at x^*, let $x_0 \in \mathcal{M}$ be an initial point and R be an upper bound on the distance $d(x_0, x^*)$. If f is L-smooth, and (possibly μ-strongly) g-convex in a closed ball of center x^* and radius $O(R)$, our algorithms obtain the same rates of convergence as AGD, up to logarithmic factors and up to a geometric penalty factor, cf. Theorem 6. See Table 1 for a succinct comparison among accelerated algorithms and their rates. This algorithm is a consequence of the general framework we design:

Riemacon: A general accelerated Riemannian scheme. We design a Riemannian accelerated inexact proximal point method that enjoys the same rates as the Euclidean accelerated proximal point method when approximating $\min_{x\in\mathcal{X}} f(x)$, up to logarithmic factors and up to a geometric penalty factor, where $f : \mathcal{N} \subset \mathcal{M} \to \mathbb{R}$ is a g-convex (or strongly g-convex) function in a compact g-convex set $\mathcal{X} \subset \mathcal{N}$, cf. Theorem 4. Note f does not need to be smooth, provided access to the inexact prox step.

For smooth functions, we show that with access to a (not necessarily accelerated) constrained linear subroutine for strongly g-convex and smooth problems, we can inexactly solve the proximal subproblem from a warm-start point to enough accuracy so it can be used in our accelerated outer loop, in the spirit of other Euclidean algorithms like Catalyst (Lin et al., 2017). After building this machinery, we show that we are able to implement an inexact ball optimization oracle, cf. (Carmon et al., 2020), as an instance of our solution. Crucially, the diameter D of this ball depends on R and the geometry only, so in particular it is independent on the condition number of f. We can use the linearly convergent algorithm in (Criscitiello and Boumal, 2021) for the implementation of the prox subroutine, and iterating the application of the ball optimization oracle leads to global accelerated convergence (Martínez-Rubio, 2020).

Importantly, our algorithms obtain acceleration without an undesirable assumption that most previous works had to make: that the iterates of the algorithm stay inside of a *pre-specified* compact set without any mechanism for enforcing or guarateeing this condition. This condition is not the same as assuming the iterates are bounded in *some* compact set, see this discussion. All methods require some constraints to bound geometric penalties but to the best of our knowledge only two previous methods are able to enforce these constraints, and they apply to the limited settings of local optimization (Criscitiello and Boumal, 2021) and constant sectional curvature manifolds (Martínez-Rubio, 2020), respectively. Techniques in the rest of papers resort to just assuming that the iterates of their algorithms are always feasible. Removing this condition in general, global, and fully accelerated methods was posed as an open question in (Kim and Yang, 2022), that we solve for a wide class of Hadamard manifolds. The difficulty of constraining problems in order to bound geometric penalties as well as the necessity of achieving this goal in order to provide full optimization guarantees with bounded geometric penalties is something that has also been noted in other kinds of Riemannian algorithms, cf. (Hosseini and Sra, 2020).

The question concerning whether there are Riemannian analogs to Nesterov's algorithm that enjoy similar rates is a question that, to the best of our knowledge, was first formulated in (Zhang and Sra, 2016). In particular, since Nesterov's AGD uses a proximal operator of a function's linearization, they ask whether there is a Riemannian analog to this operation that could be used to obtain accelerated rates in the Riemannian case. We show that, instead, a proximal step with respect to the *whole* function can be approximated efficiently in Hadamard manifolds and it can be used along with an accelerated outer loop. Previously known Riemannian proximal methods either obtain asymptotic analyses, assume exact proximal computation, or work with approximate proximal operators by using different inexactness conditions as ours, and none of them show how to implement the proximal operators or obtain accelerated proximal point methods, cf. Section 3.

Table 1: Convergence rates of related works with provable guarantees for smooth problems over uniquely geodesic manifolds. **K?** → sectional curvature?, **G?** → global algorithm?: any initial distance to a minimizer is allowed. Here L and L′ mean they are local algorithms that require initial distance $O((L/\mu)^{-3/4})$ and $O((L/\mu)^{-1/2})$, respectively. **F?** → full acceleration?: dependence on L, μ, and ε like AGD up to possibly log factors. **C?** → can enforce some constraints?: All methods require their iterates to be in some **pre-specified** compact set, but works with ✗ just assume the iterates will remain within the constraints. We use $\mathcal{W} \stackrel{\text{def}}{=} \sqrt{\frac{L}{\mu}}\log(\frac{LR^2}{\varepsilon})$. *A mild condition on the covariant derivative of the metric tensor is required, cf. Assumption 5, Appendix B.

Method	**g-convex**	**μ-st. g-cvx**	**K?**	**G?**	**F?**	**C?**
(Nesterov, 2005, AGD)	$O(\sqrt{\frac{LR^2}{\varepsilon}})$	$O(\mathcal{W})$	0	✓	✓	✓
(Zhang and Sra, 2018)	-	$O(\mathcal{W})$	bounded	L	✓	✗
(Ahn and Sra, 2020)	-	$\widetilde{O}(\frac{L}{\mu}+\mathcal{W})$	bounded	✓	✗	✗
(Martínez-Rubio, 2020)	$\widetilde{O}(\zeta^{\frac{3}{2}}\sqrt{\frac{\zeta}{\delta}+\frac{LR^2}{\delta\varepsilon}})$	$\widetilde{O}(\zeta^{\frac{3}{2}}\cdot\mathcal{W})$	ctant.$\neq 0$	✓	✓	✓
(Criscitiello and Boumal, 2021)	-	$O(\mathcal{W})$	bounded*	L′	✓	✓
(Kim and Yang, 2022)	$O(\zeta\sqrt{\frac{LR^2}{\varepsilon}})$	$O(\zeta\cdot\mathcal{W})$	bounded	✓	✓	✗
Theorem 6	$\widetilde{O}(\zeta^2\sqrt{\zeta+\frac{LR^2}{\varepsilon}})$	$\widetilde{O}(\zeta^2\cdot\mathcal{W})$	Hadamard*	✓	✓	✓

1.1. Preliminaries

We provide definitions of Riemannian geometry concepts that we use in this work. The interested reader can refer to (Petersen, 2006; Bacák, 2014) for an in-depth review of this topic, but for this work the following notions will be enough. A Riemannian manifold $(\mathcal{M}, \mathfrak{g})$ is a real C^∞ manifold $\mathcal{M}$ equipped with a metric $\mathfrak{g}$, which is a smoothly varying, i.e., C^∞, inner product. For $x \in \mathcal{M}$, denote by $T_x\mathcal{M}$ the tangent space of $\mathcal{M}$ at x. For vectors $v, w \in T_x\mathcal{M}$, we denote the inner product of the metric by $\langle v, w\rangle_x$ and the norm it induces by $\|v\|_x \stackrel{\text{def}}{=} \sqrt{\langle v, v\rangle_x}$. Most of the time, the point x is known from context, in which case we write $\langle v, w\rangle$ or $\|v\|$.

A geodesic of length ℓ is a curve $\gamma : [0, \ell] \to \mathcal{M}$ of unit speed that is locally distance minimizing. A uniquely geodesic space is a space such that for every two points there is one and only one geodesic that joins them. In such a case the exponential map $\operatorname{Exp}_x : T_x\mathcal{M} \to \mathcal{M}$ and the inverse exponential map $\operatorname{Log}_x : \mathcal{M} \to T_x\mathcal{M}$ are well defined for every pair of points, and are as follows. Given $x, y \in \mathcal{M}$, $v \in T_x\mathcal{M}$, and a geodesic γ of length $\|v\|$ such that $\gamma(0) = x$, $\gamma(\|v\|) = y$, $\gamma'(0) = v/\|v\|$, we have that $\operatorname{Exp}_x(v) = y$ and $\operatorname{Log}_x(y) = v$. We denote by $d(x, y)$ the distance between x and y, and note that it takes the same value as $\|\operatorname{Log}_x(y)\|$. The manifold $\mathcal{M}$ comes with a natural parallel transport of vectors between tangent spaces, that formally is defined from a way of identifying nearby tangent spaces, known as the Levi-Civita connection ∇ (Levi-Civita, 1977). We use this parallel transport throughout this work. As all previous accelerated related works do, discussed in Section 3, we assume that we can compute the exponential and inverse exponential maps, and parallel transport of vectors for our manifold.

Given a 2-dimensional subspace $V \subseteq T_x\mathcal{M}$ of the tangent space of a point x, the sectional curvature at x with respect to V is defined as the Gauss curvature, for the surface $\mathrm{Exp}_x(V)$ at x. The Gauss curvature at a point x can be defined as the product of the maximum and minimum curvatures of the curves resulting from intersecting the surface with planes that are normal to the surface at x. A Hadamard manifold is a complete simply connected Riemannian manifold whose sectional curvature is non-positive, like the hyperbolic space or the space of $n\times n$ symmetric positive definite matrices with the metric $\langle X, Y\rangle_A \stackrel{\mathrm{def}}{=} \mathrm{Tr}(A^{-1}XA^{-1}Y)$ where X, Y are in the tangent space of A. Hadamard manifolds are uniquely geodesic. Note that in a general manifold $\mathrm{Exp}_x(\cdot)$ might not be defined for each $v \in T_x\mathcal{M}$, but in a Hadamard manifold of dimension n, the exponential map at any point is a global diffeomorphism between $T_x\mathcal{M} \cong \mathbb{R}^n$ and the manifold, and so the exponential map is defined everywhere. We now proceed to define the main properties that will be assumed on our model for the function to be minimized and on the feasible set $\mathcal{X}$.

Definition 1 (Geodesic Convexity and Smoothness) *Let $f : \mathcal{N} \subset \mathcal{M} \to \mathbb{R}$ be a differentiable function defined on an open set $\mathcal{N}$ contained in a Riemannian manifold $\mathcal{M}$. Given $L \geq \mu > 0$, we say that f is* L-smooth *in a set $\mathcal{X} \subseteq \mathcal{N}$ if for any two points $x, y \in \mathcal{X}$, f satisfies*

$$f(y) \leq f(x) + \langle \nabla f(x), \mathrm{Log}_x(y)\rangle + \frac{L}{2}d(x, y)^2.$$

Analogously, we say that f is μ-strongly g-convex *in $\mathcal{X}$, if for any two points $x, y \in \mathcal{X}$, we have*

$$f(y) \geq f(x) + \langle \nabla f(x), \mathrm{Log}_x(y)\rangle + \frac{\mu}{2}d(x, y)^2.$$

If the previous inequality is satisfied with $\mu = 0$, we say the function is g-convex *in $\mathcal{X}$. If f is not differentiable, we say f is μ-strongly g-convex in $\mathcal{X}$ if for all $x, y \in \mathcal{X}$, and $t \in [0, 1]$:*

$$f(\mathrm{Exp}_x(t \cdot \mathrm{Log}_x(x) + (1-t) \cdot \mathrm{Log}_x(y))) \leq tf(x) + (1-t)f(y) - \frac{t(1-t)\mu}{2}d(x, y)^2.$$

Again, if the inequality is satisfied for $\mu = 0$, we have g-convexity in $\mathcal{X}$. This definition coincides with the previous one when f is differentiable.

We present the following fact about the squared-distance function, when one of the arguments is fixed. The constants ζ_D, δ_D below appear everywhere in Riemannian first-order optimization methods because, among other things, Fact 2 yields Riemannian inequalities that are analogous to the equality in the Euclidean cosine law of a triangle, cf. Corollary 15, and these inequalities have wide applicability in the analyses of Riemannian methods.

Fact 2 (Local information of the squared-distance) *Let $\mathcal{M}$ be a Riemannian manifold of sectional curvature bounded by $[\kappa_{\min}, \kappa_{\max}]$ that contains a uniquely g-convex set $\mathcal{X} \subset \mathcal{M}$ of diameter $D < \infty$. Then, given $x, y \in \mathcal{X}$ we have the following for the function $\Phi_x : \mathcal{M} \to \mathbb{R}$, $y \mapsto \frac{1}{2}d(x, y)^2$:*

$$\nabla\Phi_x(y) = -\mathrm{Log}_y(x),$$

and

$$\delta_D\|v\|^2 \leq \mathrm{Hess}\,\Phi_x(y)[v, v] \leq \zeta_D\|v\|^2.$$

These bounds are tight for spaces of constant sectional curvature. The geometric constants are

$$\zeta_D \stackrel{\text{def}}{=} \begin{cases} D\sqrt{|\kappa_{\min}|}\coth(D\sqrt{|\kappa_{\min}|}) & \text{if } \kappa_{\min} \leq 0 \\ 1 & \text{if } \kappa_{\min} > 0 \end{cases},$$

and

$$\delta_D \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } \kappa_{\max} \leq 0 \\ D\sqrt{\kappa_{\max}}\cot(D\sqrt{\kappa_{\max}}) & \text{if } \kappa_{\max} > 0 \end{cases}.$$

Consequently, Φ_x is δ_D-strongly g-convex and ζ_D-smooth in $\mathcal{X}$. See (Kim and Yang, 2022), for instance. In particular, for Hadamard manifolds, Φ_x is 1-strongly g-convex and sublevel sets of g-convex functions are g-convex sets, so balls are g-convex in these manifolds (Bacák, 2014).

1.2. Notation

Let $\mathcal{M}$ be a uniquely geodesic n-dimensional Riemannian manifold. Given points $x, y, z \in \mathcal{M}$, we abuse the notation and write y in non-ambiguous and well-defined contexts in which we should write $\operatorname{Log}_x(y)$. For example, for $v \in T_x\mathcal{M}$ we have $\langle v, y - x\rangle = -\langle v, x - y\rangle = \langle v, \operatorname{Log}_x(y) - \operatorname{Log}_x(x)\rangle = \langle v, \operatorname{Log}_x(y)\rangle$; $\|v - y\| = \|v - \operatorname{Log}_x(y)\|$; $\|z - y\|_x = \|\operatorname{Log}_x(z) - \operatorname{Log}_x(y)\|$; and $\|y - x\|_x = \|\operatorname{Log}_x(y)\| = d(y, x)$. We denote by $\mathcal{X}$ a compact, uniquely geodesic g-convex set of diameter D contained in an open set $\mathcal{N} \subset \mathcal{M}$ and we use $I_{\mathcal{X}}$ for the indicator function of $\mathcal{X}$, which is 0 at points in $\mathcal{X}$ and $+\infty$ otherwise. For a vector $v \in T_y\mathcal{M}$, we use $\Gamma_y^x(v) \in T_x\mathcal{M}$ to denote the parallel transport of v from $T_y\mathcal{M}$ to $T_x\mathcal{M}$ along the unique geodesic that connects y to x. We call $f : \mathcal{N} \subset \mathcal{M} \to \mathbb{R}$ a g-convex function we want to optimize. We use ε to denote the approximation accuracy parameter, $x_0 \in \mathcal{X}$ for the initial point of our algorithms, and $\bar{R} \stackrel{\text{def}}{=} d(x_0, \bar{x}^*)$ for the initial distance to an arbitrary constrained minimizer $\bar{x}^* \in \arg\min_{x\in\mathcal{X}} f(x)$. We use R for an upper bound on the initial distance $d(x_0, x^*)$ to an unconstrained minimizer x^*, if it exists. The big-O notation $\widetilde{O}(\cdot)$ omits $\log$ factors. Note that in the setting of Hadamard manifolds, the bounds on the sectional curvature are $\kappa_{\min} \leq \kappa_{\max} \leq 0$. Hence for notational convenience, we define $\bar{\zeta} \stackrel{\text{def}}{=} \zeta_D = D\sqrt{|\kappa_{\min}|}\coth(D\sqrt{|\kappa_{\min}|}) \geq 1$, $\bar{\delta} \stackrel{\text{def}}{=} 1$, and similarly $\zeta \stackrel{\text{def}}{=} \zeta_R$ and $\delta \stackrel{\text{def}}{=} \delta_R = 1$. If $v \in T_x\mathcal{M}$, we use $\Pi_{\bar{B}(0,r)}(v) \in T_x\mathcal{M}$ for the projection of v onto the closed ball with center at 0 and radius r.

2. Algorithmic framework and convergence results

In this section, we present our **Rieman**nian **ac**celerated algorithm for **con**strained g-convex optimization, or Riemacon[1]. This is a general framework that we later instantiate to provide a full algorithm. Recall our abuse of notation for points $p \in \mathcal{M}$ to mean $\operatorname{Log}_q(p)$ in contexts in which one should place a vector in $T_q\mathcal{M}$ and note that in our algorithm x_k and y_k are points in $\mathcal{M}$ whereas $z_k^{x_k} \in T_{x_k}\mathcal{M}$, $z_k^{y_k}, \bar{z}_k^{y_k} \in T_{y_k}\mathcal{M}$.

We start with an interpretation of our algorithm that helps understanding its high-level ideas. The following intends to be a qualitative explanation, and we refer to the pseudocode and the appendix for the exact descriptions and analysis. Euclidean accelerated algorithms can be interpreted, cf. (Allen Zhu and Orecchia, 2017), as a combination of a gradient descent (GD) algorithm and an online learning algorithm with losses being the affine lower bounds $f(x_k) + \langle \nabla f(x_k), \cdot - x_k\rangle$ we

1. Riemacon rhymes with “rima con” in Spanish.

obtain on $f(\cdot)$ by applying convexity at some points x_k. That is, the latter builds a lower bound estimation on f. By selecting the next query to the gradient oracle as a cleverly picked convex combination of the predictions given by these two algorithms, one can show that the instantaneous regret of the online learning algorithm can be compensated by the local progress GD makes, up to a difference of potential functions, which leads to accelerated convergence. In Riemannian optimization, there are two main obstacles. Firstly, the first-order approximations of f at points x_k yield functions that are affine but only with respect to their respective $T_{x_k}\mathcal{M}$, and so combining these lower bounds that are only simple in their tangent spaces makes obtaining good global estimations difficult. Secondly, when one obtains such global estimations, then one naturally incurs an instantaneous regret that is worse by a factor than is usual in Euclidean acceleration. This factor is a geometric constant depending on the diameter D of a set $\mathcal{X}$ where the iterates and a (possibly constrained) minimizer lie. As a consequence, the learning rate of GD would need to be multiplicatively increased by such a constant with respect to the one of the online learning algorithm in order for the regret to still be compensated with the local progress of GD (and the rates worsen by this constant). But if we fix some $\mathcal{X}$ of finite diameter, because GD's learning rate is now larger, it is not clear how to keep the iterates in $\mathcal{X}$. And if we do not have the iterates in one such set $\mathcal{X}$, then our geometric penalties could grow arbitrarily.

We find the answer in implicit methods. An implicit Euclidean (sub)gradient descent step is one that computes, from a point $x_k \in \mathcal{X}$, another point $y_k^* = x_k - \lambda v_k \in \mathcal{X}$, where $v_k \in \partial(f + I_{\mathcal{X}})(y_k^*)$, is a subgradient of $f + I_{\mathcal{X}}$ at y_k^*. Intuitively, if we could implement a Riemannian version of an implicit GD step then it should be possible to still compensate the regret of the other algorithm and keep all the iterates in the set $\mathcal{X}$. Computing such an implicit step is computationally hard in general, but we show that approximating the proximal objective $h_k(y) \stackrel{\text{def}}{=} f(y) + \frac{1}{2\lambda}d(x_k, y)^2$ with enough accuracy yields an approximate subgradient that can be used to obtain an accelerated algorithm as well. In particular, we provide an accelerated scheme for which we show that the error incurred by the approximation of the subgradient can be bounded by some terms we can control, cf. Lemma 8, namely a small term that appears in our Lyapunov function and also a term proportional to the squared norm of the approximated subgradient, which only increases the final convergence rates by a constant. For L-smooth functions, we show that an unaccelerated linearly convergent subroutine initialized at the warm-started point achieves the desired accuracy of the subproblem fast, cf. Appendix B. This proximal approach works by exploiting the fact that the Riemannian Moreau envelop is g-convex in Hadamard manifolds (Azagra and Ferrera, 2005) and that the subproblem h_k, defined with $\lambda = \zeta_{2D}/L$, is strongly g-convex and smooth with a condition number that only depends on the geometry. For this reason, a local algorithm like the one in (Criscitiello and Boumal, 2021) can be implemented in balls whose radius is independent on the condition number of f. Besides these steps, we use a coupling of the approximate implicit RGD and of a mirror descent (MD) algorithm, along with a technique in (Kim and Yang, 2022) to move dual points to the right tangent spaces without incurring extra geometric penalties, that we adapt to work with dual projections, cf. Lemma 9. Importantly, the MD algorithm keeps the dual point close to the set $\mathcal{X}$ by using the projection in Line 12, which implies that the point x_k is close to $\mathcal{X}$ as well, and this is crucial to keep low geometric penalties. This MD approach is a mix between follow-the-regularized-leader algorithms, that do not project the dual variable, and pure mirror descent algorithms that always project the dual variable. In the analysis, we note that partial projection also works, meaning that defining a new dual point that is closer to all of the points in the feasible set but without being a full projection leads to the same guarantees. Because we use the

Algorithm 1 Riemacon: **Riema**nnian **Ac**celeration - **Con**strained g-Convex Optimization

Input: Feasible set $\mathcal{X}$. Initial point $x_0 \in \mathcal{X} \subset \mathcal{N}$. Function $f : \mathcal{N} \subset \mathcal{M} \to \mathbb{R}$ that is g-convex in $\mathcal{X}$, for a Hadamard manifold $\mathcal{M}$. Parameter $\lambda > 0$. Optionally: final iteration T or accuracy ε. If ε is provided, compute the corresponding T, cf. Theorem 4.

Parameters:

- Geometric penalty $\xi \stackrel{\text{def}}{=} 4\zeta_{2D} - 3 \leq 8\bar{\zeta} - 3 = O(\bar{\zeta})$.
- Implicit Gradient Descent learning rate λ.
- Mirror Descent learning rates $\eta_k \stackrel{\text{def}}{=} a_k/\xi$.
- Proportionality constant in the proximal subproblem accuracies: $\Delta_k \stackrel{\text{def}}{=} \frac{1}{(k+1)^2}$.

Definition: (computation of this value is not needed)

- Prox. accuracies: $\sigma_k \stackrel{\text{def}}{=} \frac{\Delta_k d(x_k, y_k^*)^2}{78\lambda}$ where $y_k^* \stackrel{\text{def}}{=} \arg\min_{y \in \mathcal{X}}\{f(y) + \frac{1}{2\lambda} d(x_k, y)^2\}$.

$y_0 \leftarrow \sigma_0$-minimizer of the proximal problem $\min_{y\in\mathcal{X}}\{f(y) + \frac{1}{2\lambda}d(x_0, y)^2\}$
$\bar{z}_0^{y_0} \leftarrow z_0^{y_0} \leftarrow 0 \in T_{y_0}\mathcal{M}$
$A_0 \leftarrow 200\lambda\xi$
for $k = 1$ **to** T **do**
  $a_k \leftarrow 2\lambda \frac{k+32\xi}{5}$
  $A_k \leftarrow a_k/\xi + A_{k-1} = \sum_{i=1}^k a_i/\xi + A_0 = \lambda\left(\frac{k(k+1+64\xi)}{5\xi} + 200\xi\right)$
  $x_k \leftarrow \operatorname{Exp}_{y_{k-1}}(\frac{a_k}{A_{k-1}+a_k}\bar{z}_{k-1}^{y_{k-1}} + \frac{A_{k-1}}{A_{k-1}+a_k}y_{k-1}) = \operatorname{Exp}_{y_{k-1}}(\frac{a_k}{A_{k-1}+a_k}\bar{z}_{k-1}^{y_{k-1}})$ ◇ Coupling
  $y_k \leftarrow \sigma_k$-minimizer of $y \mapsto \{f(y) + \frac{1}{2\lambda}d(x_k, y)^2\}$ in $\mathcal{X}$ ◇ Approximate implicit RGD
  $v_k^x \leftarrow -\operatorname{Log}_{x_k}(y_k)/\lambda$ ◇ Approximate subgradient
  $z_k^{x_k} \leftarrow \operatorname{Log}_{x_k}(\operatorname{Exp}_{y_{k-1}}(\bar{z}_{k-1}^{y_{k-1}})) - \eta_k v_k^x$ ◇ Mirror Descent step
  $z_k^{y_k} \leftarrow \Gamma_{x_k}^{y_k}(z_k^{x_k}) + \operatorname{Log}_{y_k}(x_k)$ ◇ Moving the dual point to $T_{y_k}\mathcal{M}$
  $\bar{z}_k^{y_k} \leftarrow \Pi_{\bar{B}(0,D)}(z_k^{y_k}) \in T_{y_k}\mathcal{M}$ ◇ Easy projection done so the dual point is not very far
end for
return y_T.

mirror descent lemma over $T_{y_k}\mathcal{M}$, what we described translates to: we can project the dual $z_k^{y_k}$ onto a ball defined on $T_{y_k}\mathcal{M}$ that contains the pulled-back set $\operatorname{Log}_{y_k}(\mathcal{X})$ and by means of that trick we can keep the iterates x_k close to $\mathcal{X}$. And at the same time, the point for which we prove guarantees, namely y_k, is always in $\mathcal{X}$.

Finally, under L-smoothness, we instantiate our subroutine with the algorithm in (Criscitiello and Boumal, 2021), in balls of radius independent on the condition number of f and conclude in Theorem 6 that if we iterate this approximate implementation of a ball optimization oracle, we obtain convergence at a globally accelerated rate. We note (Zhang and Sra, 2016, Thm. 15) provided a claimed linearly convergent algorithm for constrained strongly g-convex smooth problems, and thus in principle it could be used for our subroutine after a warm start. Unfortunately, we noticed that the proof is flawed when the optimization is constrained. The first inequality in their proof only holds in general for unconstrained problems and not for projected Riemannian gradient descent, not even for the Euclidean constrained case.

We leave the proofs of most of our results to the appendix and state our main theorems below. Using the insights explained above, we show the following inequality on ψ_k, defined below, that will be used as a Lyapunov function to prove the convergence rates of Algorithm 1.

Proposition 3 [↓] *By using the notation of Algorithm 1, let*

$$\psi_k \stackrel{\text{def}}{=} A_k(f(y_k) - f(\bar{x}^*)) + \frac{1}{2}\|z_k^{y_k} - \operatorname{Log}_{y_k}(\bar{x}^*)\|_{y_k}^2 + \frac{\xi - 1}{2}\|z_k^{y_k}\|_{y_k}^2.$$

Then, for all $k \geq 1$, we have $(1 - \Delta_k)\psi_k \leq \psi_{k-1}$.

With this proposition, we can show the convergence of Riemacon for g-convex functions.

Theorem 4 [↓] *Let $\mathcal{M}$ be a finite-dimensional Hadamard manifold of bounded sectional curvature, and consider $f : \mathcal{N} \subset \mathcal{M} \to \mathbb{R}$ be a g-convex function in a compact g-convex set $\mathcal{X} \subset \mathcal{N}$ of diameter D, $\lambda \in \mathbb{R}_{>0}$, $\bar{x}^* \in \arg\min_{x \in \mathcal{X}} f(x)$, and $\bar{R} \stackrel{\text{def}}{=} d(x_0, \bar{x}^*)$. For any $\varepsilon > 0$, Algorithm 1 yields an ε-minimizer $y_T \in \mathcal{X}$ after $T = O(\bar{\zeta}\sqrt{\frac{\bar{R}^2}{\lambda\varepsilon}})$ iterations. If the function is μ-strongly g-convex then, via a sequence of restarts, we converge in $O((\bar{\zeta}\sqrt{\frac{1}{\lambda\mu}} + 1)\log(\frac{\mu\bar{R}^2}{\varepsilon}))$ iterations.*

We note that a straightforward corollary from our results is that if we can compute the exact Riemannian proximal point operator and we use it as the implicit gradient descent step in Line 8 of Algorithm 1, then the method is an accelerated proximal point method. One such Riemannian algorithm was unknown in the literature as well. Note we do not require smoothness of f.

Finally, we instantiate Algorithm 1 to implement approximate ball optimization oracles in an accelerated way. Applying these oracles sequentially leads to global accelerated convergence (Martínez-Rubio, 2020). Moreover, we show that the iterates do not get farther than $2R$ from x^*, which ultimately leads to the geometric penalty being a function of ζ and not on the condition number of f. For the subroutine in Line 8 of Algorithm 1, we use the algorithm in (Criscitiello and Boumal, 2021, Section 6), and for that we require the following.

Assumption 5 *Let $\mathfrak{R}$ be the curvature tensor of a Riemannian manifold $\mathcal{M}$. Its covariant derivative is $\nabla\mathfrak{R} = 0$.*

Locally symmetric manifolds, like the SPD matrix manifold, manifolds of constant sectional curvature, $\mathrm{SO}(n)$, the Grasmannian manifold, are all manifolds such that $\nabla\mathfrak{R} = 0$. We argue that this assumption is mild, since in particular these manifolds cover all of the applications in Section 1.

Theorem 6 [↓] *Let $\mathcal{M}$ be a finite-dimensional Hadamard manifold of bounded sectional curvature satisfying Assumption 5. Consider $f : \mathcal{N} \subset \mathcal{M} \to \mathbb{R}$ be an L-smooth and μ-strongly g-convex differentiable function in $\bar{B}(x^*, 3R)$, where x^* is its global minimizer and where $R \geq d(x_0, x^*)$ for an initial point x_0. For any $\varepsilon > 0$, Algorithm 2 yields an ε-minimizer after $\widetilde{O}(\zeta^2\sqrt{L/\mu}\log(LR^2/\varepsilon))$ calls to the gradient oracle of f. By using regularization, this algorithm ε-minimizes the g-convex case ($\mu = 0$) after $\widetilde{O}(\zeta^2\sqrt{\zeta + LR^2/\varepsilon})$ gradient oracle calls.*

In sum, the algorithm enjoys the same rates as AGD in the Euclidean space up to a factor of $\zeta^2 = R^2|\kappa_{\min}|\coth^2(R\sqrt{|\kappa_{\min}|}) \leq (1 + R \cdot |\kappa_{\min}|)^2$ (our geometric penalty) and up to universal constants and log factors. Note that as the minimum curvature $\kappa_{\min}$ approaches 0 we have $\zeta \to 1$.

Algorithm 2 Boosted Riemacon: ball optimization boosting of a Riemacon instance (Algorithm 1)

Input: Differentiable function $f : \mathcal{N} \subset \mathcal{M} \to \mathbb{R}$ that is L-smooth and μ-strongly g-convex in $\bar{B}(x^*, 3R) \subset \mathcal{N}$; initial point $x_0 \in \mathcal{N}$; bound $R \geq d(x_0, x^*)$; accuracy ε.
RiemaconSC: The strongly convex version of Algorithm 1 in Theorem 4 (cf. its proof) using (Criscitiello and Boumal, 2021) as subroutine.

if $2R \leq (46R|\kappa_{\min}|\zeta_{2R})^{-1}$ **then**
  return RiemaconSC$(\bar{B}(x_0, R), x_0, f, \zeta_{4R}/L, \varepsilon)$
end if
Compute D such that $D = (46R|\kappa_{\min}|\zeta_D)^{-1}$. Alternatively, make $D \leftarrow (70R|\kappa_{\min}|)^{-1}$.
$T \leftarrow \lceil \frac{4R}{D} \ln(\frac{LR^2}{\varepsilon}) \rceil$
$\varepsilon' \leftarrow \min\{\frac{D\varepsilon}{8R}, \frac{\mu R^2}{2T^2}\}$
$\lambda \leftarrow \zeta_{2D}/L$
for $k = 1$ **to** T **do**
  $\mathcal{X}_k \leftarrow \bar{B}(x_{k-1}, D/2)$
  $x_k \leftarrow$ RiemaconSC$(\mathcal{X}_k, x_{k-1}, f, \lambda, \varepsilon')$
end for
return x_T.

Also, we emphasize that our algorithms only needs to query the gradient of f at points in $\mathcal{X}$ and the L-smoothness and μ-strong g-convexity of f only need to hold in $\mathcal{X}$. This is relevant because in Riemannian manifolds the condition number L/μ can have a lower bound depending on the size of the set, cf. (Martínez-Rubio, 2020, Proposition 28). Intuitively, although there are twice differentiable functions defined over the Euclidean space whose Hessian is constant everywhere, in other Riemannian cases the metric may preclude having such global condition and the larger the set is, the larger the minimum possible condition number becomes. Compare this, for instance, with the bounds on the Hessian's eigenvalues of the squared-distance function in Fact 2.

3. Related work and comparisons

We compare our results with previous works. We have summarized most of the following discussion in Table 1. We include Nesterov's AGD in the table for comparison purposes[2]. There are some works on Riemannian acceleration that focus on empirical evaluation or that work under strong assumptions (Liu et al., 2017; Alimisis et al., 2019; Huang and Wei, 2019a; Alimisis et al., 2020; Lin et al., 2020), see (Martínez-Rubio, 2020) for instance for a discussion on these works. We focus the discussion on the most related work with guarantees. (Zhang and Sra, 2018) obtain an algorithm that, up to constants, achieves the same rates as AGD in the Euclidean space, for L-smooth and μ-strongly g-convex functions but only *locally*, namely when the initial point starts in a small neighborhood N of the minimizer x^*: a ball of radius $O((\mu/L)^{3/4})$ around it. (Ahn and Sra, 2020) generalize the previous algorithm and, by using similar ideas as in (Zhang and Sra, 2018) for estimating a lower bound on f, they adapt the algorithm to work globally, proving that it eventually decreases the objective as fast as AGD. However, as (Martínez-Rubio, 2020) noted, it takes as many

2. Note that the original method in (Nesterov, 1983) needed to query the gradient of the function outside of the feasible set, and this was later improved to only require queries at feasible points (Nesterov, 2005) as in our work, hence our choice of citation in Table 1.

iterations as the ones needed by Riemannian gradient descent (RGD) to reach the neighborhood of the previous algorithm. The latter work also noted that in fact RGD and the algorithm in (Zhang and Sra, 2018) can be run in parallel and combined to obtain the same convergence rates as in (Ahn and Sra, 2020), which suggested that for this technique, full acceleration with the rates of AGD only happens over the small neighborhood N in (Zhang and Sra, 2018). Note however that (Ahn and Sra, 2020) show that their algorithm will decrease the function value faster than RGD, but this is not quantified. (Jin and Sra, 2021) developed a different framework, arising from (Ahn and Sra, 2020) but with the same guarantees for accelerated first-order methods. We do not feature it in the table. (Criscitiello and Boumal, 2021) showed, under mild assumptions, that in a ball of center $x \in \mathcal{M}$ and radius $O((\mu/L)^{1/2})$ containing x^*, the pullback function $f \circ \operatorname{Exp}_x : T_x\mathcal{M} \to \mathbb{R}$ is Euclidean, strongly convex, and smooth with condition number $O(L/\mu)$, so AGD yields local acceleration as well. In short, acceleration is possible in a small neighborhood because there the manifold is almost Euclidean and the geometric deformations are small in comparison to the curvature of the objective. These techniques fail for the g-convex case since the neighborhood becomes a point ($\mu/L = 0$).

Finding fully accelerated algorithms that are *global* presents a harder challenge. By a fully accelerated algorithm we mean one with rates with same dependence as AGD on L, ε, and if it applies, on μ. Martínez-Rubio (2020) provided such algorithms for g-convex functions, strongly or not, defined over manifolds of constant sectional curvature and constrained to a ball of radius R. The convergence rates initially had large constants with respect to R but were later improved, cf. Table 1. Kim and Yang (2022) designed global algorithms with the same rates as AGD up to universal constants and a factor of $\bar{\zeta}$, their geometric penalty. However, they need to assume that the iterates of their algorithm remain in their feasible set $\mathcal{X}$ and they point out on the necessity of removing such an assumption, which they leave as an open question. Our work solves this question for a wide class of Hadamard manifolds. In their technique, they show they can use the structure of the accelerated scheme to *move* lower bound estimations oGeneralization in deep networks: The role of distance from initializationn $f(x^*)$ from one particular tangent space to another without incurring extra errors, when the right Lyapunov function is used. By *moving* lower bounds here we mean finding suitable lower bounds that are simple (a quadratic in their case), when pulled-back to one tangent space, if we start with a similar bound that is simple when pulled-back to another tangent space. We note that both our result in Theorem 4 and the results of (Kim and Yang, 2022) incur a geometric penalty ζ, when $R = D$ and for a full algorithm, our geometric penalties are ζ^2 after we factor in the implementation of the subroutine and the ball optimization oracle, cf. Theorem 6. In other words, our frameworks incur the same geometric penalties and, for a full algorithm, we implement a subroutine that adds another factor of ζ, while for a full algorithm (Kim and Yang, 2022) requires an assumption that is not clear how to satisfy. We note that better subroutines have the potential of decreasing the total geometric penalty of an instance of our framework and this is an interesting direction for future research. We also note that in machine learning applications, it has been observed that the iterates do not get far from initialization (Nagarajan and Kolter, 2019), especially in overparametrized models. In such a case, ζ can be treated as a constant.

Lower bounds. In this paragraph, we omit constants depending on the curvature bounds in the big-O notations for simplicity. (Hamilton and Moitra, 2021) proved an optimization lower bound showing that acceleration in Riemannian manifolds is harder than in the Euclidean space. (Criscitiello and Boumal, 2021) largely generalized their results. They essentially show that for a large family of Hadamard manifolds, there is a function that is smooth and strongly g-convex in a ball of radius R that contains the minimizer x^*, and for which finding a point that is $R/5$ close to x^*

requires $\widetilde{\Omega}(R)$ calls to the gradient oracle. Note that these results do not preclude the existence of a fully accelerated algorithm with rates $\widetilde{O}(R)$+AGD rates, for instance. A similar hardness statement is provided for smooth and only g-convex functions. Also, reductions as in (Martínez-Rubio, 2020) evince this hardness is also present in this case.

Handling constraints to bound geometric penalties. In our algorithm and in all other known fully accelerated algorithms, learning rates depend on the diameter of the feasible set. This is natural: estimation errors due to geometric deformations depend on the diameter via the constants ζ_D, δ_D, the cosine-law Riemannian inequalities Corollary 15, or other analogous inequalities, and the algorithms take these errors into account. All other previous works are not able to deal with any constraints and hence they simply assume that the iterates of their algorithms stay within one such pre-specified set, except for (Martínez-Rubio, 2020) and (Criscitiello and Boumal, 2021) that enforce a ball constraint, as we explained above. However, these two works have their applicability limited to spaces of constant curvature and to local optimization, respectively. Note that even if one could show that given a choice of learning rate, convergence implies that the iterates will remain in some compact set, then because the learning rates depend on the diameter of the set, and the diameter of the set would depend on the learning rates, one cannot conclude from this argument that the assumption these works make is going to be satisfied. In contrast, in this work, we design a general accelerated framework and an instance of it that keep the iterates bounded in a set we *pre-specify*, effectively bounding geometric penalties while we do not need to resort to any other extra assumptions, solving the open question in (Kim and Yang, 2022).

Some other works study and use Riemannian metric projections, see (Walter, 1974; Hosseini and Pouryayevali, 2013; Barani et al., 2013; Bacák, 2014; Zhang and Sra, 2016) and references therein. Among them, (Zhang and Sra, 2016) introduced several deterministic and stochastic first-order methods that use metric-projection oracles.

Riemannian proximal methods. There are some works that study proximal methods in Riemannian manifolds, but most of them focus on asymptotic results or assume the proximal operator can be computed exactly (Wang et al., 2015; Bento et al., 2017, 2016; Khammahawong et al., 2021; Chang et al., 2021). The rest of these works study proximal point methods under different inexact versions of the proximal operator as ours and they do not show how to implement their inexact version in applications, like in our case of smooth and g-convex optimization. In contrast, we implement the inexact proximal operator with a first-order method. (Ahmadi and Khatibzadeh, 2014) provide a convergence analysis of an inexact proximal point method but when applied to optimization they assume the computation of the proximal operator is exact. (Tang and Huang, 2014) uses a different inexact condition and proves linear convergence, under a growth condition on f. (Wang et al., 2016b) obtains linear convergence of an inexact proximal point method under a different growth assumption on f and under an absolute error condition on the proximal function. Most importantly, none of these methods presented acceleration.

4. Conclusion and future directions

In this work, we pursued an approach that, by designing and making use of inexact Riemannian proximal methods, yielded accelerated optimization algorithms. Consequently we were able to work without an undesirable assumption that most previous methods required, whose potential satisfiability is not clear: that the iterates stay in certain specified geodesically-convex set without enforcing them to be in the set. A future direction of research is the study of whether there are algorithms like ours that incur even lower geometric penalties or that do not incur $\log(1/\varepsilon)$ extra factors. Determining whether the convergence rates of fully accelerated algorithms necessarily incur a geometric factor is an interesting open problem: current lower bounds only require an additive geometric penalty and the rate of unaccelerated unconstrained RGD in (Zhang and Sra, 2016, Thm. 15) does present an additive geometric constant only, while all known accelerated methods have a multiplicative geometric constant in their rates. Note that for the local algorithms in Table 1, this factor is a constant. Another interesting direction consists of studying generalizations of our approach to more general manifolds, namely manifolds of non-negative or even of bounded sectional cuvature.

Acknowledgments

We thank Christophe Roux for proofreading and bringing an error to our attention in an early version of this draft. David Martínez-Rubio was partially funded by the DFG Cluster of Excellence MATH+ (EXC-2046/1, project id 390685689) funded by the Deutsche Forschungsgemeinschaft (DFG).

References

Parviz Ahmadi and Hadi Khatibzadeh. On the convergence of inexact proximal point algorithm on Hadamard manifolds. *Taiwanese Journal of Mathematics*, 18(2):419–433, 2014. URL *https://doi.org/10.11650/tjm.18.2014.3066*.

Kwangjun Ahn and Suvrit Sra. From Nesterov's estimate sequence to Riemannian acceleration. *arXiv preprint arXiv:2001.08876*, 2020. URL *https://arxiv.org/abs/2001.08876*.

Foivos Alimisis, Antonio Orvieto, Gary Bécigneul, and Aurelien Lucchi. A continuous-time perspective for modeling acceleration in Riemannian optimization. *arXiv preprint arXiv:1910.10782*, 2019. URL *https://arxiv.org/abs/1910.10782*.

Foivos Alimisis, Antonio Orvieto, Gary Bécigneul, and Aurelien Lucchi. Practical accelerated optimization on Riemannian manifolds. *arXiv preprint arXiv:2002.04144*, 2020. URL *https://arxiv.org/abs/2002.04144*.

Zeyuan Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. *J. Mach. Learn. Res.*, 18:221:1–221:51, 2017. URL *http://jmlr.org/papers/v18/16-410.html*.

Zeyuan Allen-Zhu. Natasha 2: Faster non-convex optimization than SGD. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 2680–2691, 2018. URL *http://papers.nips.cc/paper/7533-natasha-2-faster-non-convex-optimization-than-sgd*.

Zeyuan Allen Zhu and Lorenzo Orecchia. Nearly-linear time positive LP solver with faster convergence rate. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 229–236, 2015. doi: 10.1145/2746539.2746573. URL *https://doi.org/10.1145/2746539.2746573*.

Zeyuan Allen Zhu and Lorenzo Orecchia. Linear coupling: An ultimate unification of gradient and mirror descent. In *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*, pages 3:1–3:22, 2017. doi: 10.4230/LIPIcs.ITCS.2017.3. URL *https://doi.org/10.4230/LIPIcs.ITCS.2017.3*.

Zeyuan Allen-Zhu, Ankit Garg, Yuanzhi Li, Rafael Oliveira, and Avi Wigderson. Operator scaling via geodesically convex optimization, invariant theory and polynomial identity testing. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 172–181, 2018. URL *https://arxiv.org/abs/1804.01076*.

Daniel Azagra and Juan Ferrera. Inf-convolution and regularization of convex functions on Riemannian manifolds of nonpositive curvature. *arXiv preprint math/0505496*, 2005. URL *https://arxiv.org/abs/math/0505496*.

Miroslav Bacák. Convex analysis and optimization in Hadamard spaces. In *Convex Analysis and Optimization in Hadamard Spaces*. de Gruyter, 2014. doi: https://doi.org/10.1515/9783110361629.

A Barani, S Hosseini, and MR Pouryayevali. On the metric projection onto φ-convex subsets of Hadamard manifolds. *Revista matemática complutense*, 26(2):815–826, 2013. doi: https://doi.org/10.1007/s13163-011-0085-4.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009. doi: 10.1137/080716542. URL *https://doi.org/10.1137/080716542*.

Jonathan Bennett, Anthony Carbery, Michael Christ, and Terence Tao. The Brascamp–Lieb inequalities: finiteness, structure and extremals. *Geometric and Functional Analysis*, 17(5):1343–1415, 2008.

GC Bento, OP Ferreira, and PR Oliveira. Proximal point method for a special class of nonconvex functions on Hadamard manifolds. *Optimization*, 64(2):289–319, 2015.

Glaydston de Carvalho Bento, João Xavier da Cruz-Neto, and Paulo Roberto Oliveira. A new approach to the proximal point method: Convergence on general Riemannian manifolds. *J. Optim. Theory Appl.*, 168(3):743–755, 2016. doi: 10.1007/s10957-015-0861-2. URL *https://doi.org/10.1007/s10957-015-0861-2*.

Glaydston de Carvalho Bento, Orizon P. Ferreira, and Jefferson G. Melo. Iteration-complexity of gradient, subgradient and proximal point methods on Riemannian manifolds. *J. Optim. Theory Appl.*, 173(2):548–562, 2017. doi: 10.1007/s10957-017-1093-4. URL *https://doi.org/10.1007/s10957-017-1093-4*.

Léopold Cambier and Pierre-Antoine Absil. Robust low-rank matrix completion by Riemannian optimization. *SIAM J. Scientific Computing*, 38(5), 2016. doi: 10.1137/15M1025153. URL *https://doi.org/10.1137/15M1025153*.

Yair Carmon, John C. Duchi, Oliver Hinder, and Aaron Sidford. "Convex until proven guilty": Dimension-free acceleration of gradient descent on non-convex functions. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 654–663, 2017. URL *http://proceedings.mlr.press/v70/carmon17a.html*.

Yair Carmon, Arun Jambulapati, Qijia Jiang, Yujia Jin, Yin Tat Lee, Aaron Sidford, and Kevin Tian. Acceleration with a ball optimization oracle. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL *https://proceedings.neurips.cc/paper/2020/hash/dba4c1a117472f6aca95211285d0587e-Abstract.html*.

Glaydston de Carvalho Bento and Jefferson G. Melo. Subgradient method for convex feasibility on Riemannian manifolds. *J. Optim. Theory Appl.*, 152(3):773–785, 2012. doi: 10.1007/s10957-011-9921-4. URL *https://doi.org/10.1007/s10957-011-9921-4*.

Shih-Sen Chang, Jen-Chih Yao, M Liu, and LC Zhao. Inertial proximal point algorithm for variational inclusion in Hadamard manifolds. *Applicable Analysis*, pages 1–12, 2021. doi: https://doi.org/10.1080/00036811.2021.2016719.

Anoop Cherian and Suvrit Sra. Riemannian dictionary learning and sparse coding for positive definite matrices. *IEEE Trans. Neural Networks Learn. Syst.*, 28(12):2859–2871, 2017. doi: 10.1109/TNNLS.2016.2601307. URL *https://doi.org/10.1109/TNNLS.2016.2601307*.

Francisco Criado, David Martínez-Rubio, and Sebastian Pokutta. Fast algorithms for packing proportional fairness and its dual. *arXiv preprint arXiv:2109.03678*, 2021. URL *https://arxiv.org/abs/2109.03678*.

Chris Criscitiello and Nicolas Boumal. Efficiently escaping saddle points on manifolds. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 5985–5995, 2019. URL *http://papers.nips.cc/paper/8832-efficiently-escaping-saddle-points-on-manifolds*.

Chris Criscitiello and Nicolas Boumal. An accelerated first-order method for non-convex optimization on manifolds. *arXiv preprint arXiv:2008.02252*, 2020. URL *https://arxiv.org/abs/2008.02252*.

Christopher Criscitiello and Nicolas Boumal. Negative curvature obstructs acceleration for geodesically convex optimization, even with exact first-order oracles. *CoRR*, abs/2111.13263, 2021. URL *https://arxiv.org/abs/2111.13263*.

João Xavier da Cruz Neto, Orizon Pereira Ferreira, L. R. Lucambio Pérez, and Sándor Zoltán Németh. Convex- and monotone-transformable mathematical programming problems and a proximal-like point method. *J. Glob. Optim.*, 35(1):53–69, 2006. doi: 10.1007/s10898-005-6741-9. URL *https://doi.org/10.1007/s10898-005-6741-9*.

Jelena Diakonikolas and Lorenzo Orecchia. Accelerated extra-gradient descent: A novel accelerated first-order method. In *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, pages 23:1–23:19, 2018. doi: 10.4230/LIPIcs.ITCS.2018.23. URL *https://doi.org/10.4230/LIPIcs.ITCS.2018.23*.

Jelena Diakonikolas and Lorenzo Orecchia. The approximate duality gap technique: A unified theory of first-order methods. *SIAM Journal on Optimization*, 29(1):660–689, 2019. doi: 10.1137/18M1172314. URL *https://doi.org/10.1137/18M1172314*.

Yoel Drori and Marc Teboulle. Performance of first-order methods for smooth convex minimization: a novel approach. *Math. Program.*, 145(1-2):451–482, 2014. doi: 10.1007/s10107-013-0653-0. URL *https://doi.org/10.1007/s10107-013-0653-0*.

Alan Edelman, Tomás A. Arias, and Steven Thomas Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Analysis Applications*, 20(2):303–353, 1998. doi: 10.1137/S0895479895290954. URL *https://doi.org/10.1137/S0895479895290954*.

Alexander Gasnikov, Pavel E. Dvurechensky, Eduard A. Gorbunov, Evgeniya A. Vorontsova, Daniil Selikhanovych, César A. Uribe, Bo Jiang, Haoyue Wang, Shuzhong Zhang, Sébastien Bubeck, Qijia Jiang, Yin Tat Lee, Yuanzhi Li, and Aaron Sidford. Near optimal methods for minimizing convex functions with Lipschitz p-th derivatives. In *Conference on Learning Theory, COLT 2019, 25-28 June 2019, Phoenix, AZ, USA*, pages 1392–1393, 2019. URL *http://proceedings.mlr.press/v99/gasnikov19b.html*.

Matthieu Genicot, Wen Huang, and Nickolay T. Trendafilov. Weakly correlated sparse components with nearly orthonormal loadings. In *Geometric Science of Information - Second International Conference, GSI 2015, Palaiseau, France, October 28-30, 2015, Proceedings*, pages 484–490, 2015. doi: 10.1007/978-3-319-25040-3_52. URL *https://doi.org/10.1007/978-3-319-25040-3_52*.

Linus Hamilton and Ankur Moitra. A no-go theorem for acceleration in the hyperbolic plane. *arXiv preprint arXiv:2101.05657*, 2021. URL *https://arxiv.org/abs/2101.05657*.

Gennadij Heidel and Volker Schulz. A Riemannian trust-region method for low-rank tensor completion. *Numerical Lin. Alg. with Applic.*, 25(6), 2018. doi: 10.1002/nla.2175. URL *https://doi.org/10.1002/nla.2175*.

Oliver Hinder, Aaron Sidford, and Nimit Sharad Sohoni. Near-optimal methods for minimizing star-convex functions and beyond. *CoRR*, abs/1906.11985, 2019. URL *http://arxiv.org/abs/1906.11985*.

Reshad Hosseini and Suvrit Sra. Matrix manifold optimization for Gaussian mixtures. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 910–918, 2015. URL *http://papers.nips.cc/paper/5812-matrix-manifold-optimization-for-gaussian-mixtures*.

Reshad Hosseini and Suvrit Sra. An alternative to EM for gaussian mixture models: Batch and stochastic Riemannian optimization. *CoRR*, abs/1706.03267, 2017. URL *http://arxiv.org/abs/1706.03267*.

Reshad Hosseini and Suvrit Sra. Recent advances in stochastic Riemannian optimization. *Handbook of Variational Methods for Nonlinear Geometric Data*, pages 527–554, 2020.

Seyedehsomayeh Hosseini and Mohamad Pouryayevali. On the metric projection onto prox-regular subsets of Riemannian manifolds. *Proceedings of the American Mathematical Society*, 141(1): 233–244, 2013. URL *https://www.jstor.org/stable/23558396*.

Wen Huang and Ke Wei. Extending FISTA to Riemannian optimization for sparse PCA. *arXiv preprint arXiv:1909.05485*, 2019a. URL *https://arxiv.org/abs/1909.05485*.

Wen Huang and Ke Wei. Riemannian proximal gradient methods. *arXiv preprint arXiv:1909.06065*, 2019b. URL *https://arxiv.org/abs/1909.06065*.

Anastasiya Ivanova, Dmitry Pasechnyuk, Dmitry Grishchenko, Egor Shulgin, Alexander V. Gasnikov, and Vladislav Matyukhin. Adaptive catalyst for smooth convex optimization. In Nicholas N. Olenev, Yuri G. Evtushenko, Milojica Jacimovic, Michael Yu. Khachay, and Vlasta

Malkova, editors, *Optimization and Applications - 12th International Conference, OPTIMA 2021, Petrovac, Montenegro, September 27 - October 1, 2021, Proceedings*, volume 13078 of *Lecture Notes in Computer Science*, pages 20–37. Springer, 2021. doi: 10.1007/978-3-030-91059-4_2. URL *https://doi.org/10.1007/978-3-030-91059-4_2*.

Jikai Jin and Suvrit Sra. A Riemannian accelerated proximal extragradient framework and its implications. *CoRR*, abs/2111.02763, 2021. URL *https://arxiv.org/abs/2111.02763*.

Ian T Jolliffe, Nickolay T Trendafilov, and Mudassir Uddin. A modified principal component technique based on the lasso. *Journal of computational and Graphical Statistics*, 12(3):531–547, 2003. URL *https://doi.org/10.1198/1061860032148*.

Michael I. Jordan, Tianyi Lin, and Emmanouil-Vasileios Vlatakis-Gkaragkounis. First-order algorithms for min-max optimization in geodesic metric spaces. *CoRR*, abs/2206.02041, 2022. doi: 10.48550/arXiv.2206.02041. URL *https://doi.org/10.48550/arXiv.2206.02041*.

Pooria Joulani, Anant Raj, András György, and Csaba Szepesvári. A simpler approach to accelerated optimization: iterative averaging meets optimism. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 4984–4993. PMLR, 2020. URL *http://proceedings.mlr.press/v119/joulani20a.html*.

Dmitry Kamzolov and Alexander Gasnikov. Near-optimal hyperfast second-order method for convex optimization and its sliding. *arXiv preprint arXiv:2002.09050*, 2020. URL *https://arxiv.org/abs/2002.09050*.

Hiroyuki Kasai, Pratik Jawanpuria, and Bamdev Mishra. Riemannian adaptive stochastic gradient algorithms on matrix manifolds. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 3262–3271, 2019. URL *http://proceedings.mlr.press/v97/kasai19a.html*.

Konrawut Khammahawong, Poom Kumam, Parin Chaipunya, and Juan Martínez-Moreno. Tseng's methods for inclusion problems on Hadamard manifolds. *Optimization*, pages 1–35, 2021. URL *https://doi.org/10.1080/02331934.2021.1940179*.

Masoud Badiei Khuzani and Na Li. Stochastic primal-dual method on Riemannian manifolds of bounded sectional curvature. In *16th IEEE International Conference on Machine Learning and Applications, ICMLA 2017, Cancun, Mexico, December 18-21, 2017*, pages 133–140, 2017. doi: 10.1109/ICMLA.2017.0-167. URL *https://doi.org/10.1109/ICMLA.2017.0-167*.

Jungbin Kim and Insoon Yang. Accelerated gradient methods for geodesically convex optimization: Tractable algorithms and convergence analysis. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 11255–11282. PMLR, 2022. URL *https://proceedings.mlr.press/v162/kim22k.html*.

Tullio Levi-Civita. *The absolute differential calculus (calculus of tensors)*. Courier Corporation, 1977. ISBN 978-0-486-31625-3.

Mario Lezcano-Casado and David Martínez-Rubio. Cheap orthogonal constraints in neural networks: A simple parametrization of the orthogonal and unitary group. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 3794–3803, 2019. URL *http://proceedings.mlr.press/v97/lezcano-casado19a.html*.

Hongzhou Lin, Julien Mairal, and Zaïd Harchaoui. Catalyst acceleration for first-order convex optimization: from theory to practice. *J. Mach. Learn. Res.*, 18:212:1–212:54, 2017. URL *http://jmlr.org/papers/v18/17-748.html*.

Lizhen Lin, Bayan Saparbayeva, Michael Minyi Zhang, and David B. Dunson. Accelerated algorithms for convex and non-convex optimization on manifolds. *CoRR*, abs/2010.08908, 2020. URL *https://arxiv.org/abs/2010.08908*.

Yuanyuan Liu, Fanhua Shang, James Cheng, Hong Cheng, and Licheng Jiao. Accelerated first-order methods for geodesically convex optimization on Riemannian manifolds. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 4868–4877, 2017. URL *https://papers.nips.cc/paper/2017/hash/6ef80bb237adf4b6f77d0700e1255907-Abstract.html*.

David Martínez-Rubio. Global Riemannian acceleration in hyperbolic and spherical spaces. *arXiv preprint arXiv:2012.03618*, 2020. URL *https://arxiv.org/abs/2012.03618*.

Bamdev Mishra and Rodolphe Sepulchre. R3MC: A Riemannian three-factor algorithm for low-rank matrix completion. In *53rd IEEE Conference on Decision and Control, CDC 2014, Los Angeles, CA, USA, December 15-17, 2014*, pages 1137–1142, 2014. doi: 10.1109/CDC.2014.7039534. URL *https://doi.org/10.1109/CDC.2014.7039534*.

Vaishnavh Nagarajan and J. Zico Kolter. Generalization in deep networks: The role of distance from initialization. *CoRR*, abs/1901.01672, 2019. URL *http://arxiv.org/abs/1901.01672*.

Yurii Nesterov. A method of solving a convex programming problem with convergence rate o(1/k2). In *Soviet Mathematics Doklady*, volume 27, pages 372–376, 1983. URL *http://www.mathnet.ru/links/97a4ab44817d9e795c45e0d5e8a46d64/dan46009.pdf*.

Yurii Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103(1):127–152, 2005. doi: 10.1007/s10107-004-0552-5. URL *https://doi.org/10.1007/s10107-004-0552-5*.

Peter Petersen. *Riemannian geometry*, volume 171. Springer, 2006. ISBN 978-0-387-29403-2.

Hiroyuki Sato, Hiroyuki Kasai, and Bamdev Mishra. Riemannian stochastic variance reduced gradient. *CoRR*, abs/1702.05594, 2017. URL *http://arxiv.org/abs/1702.05594*.

Hiroyuki Sato, Hiroyuki Kasai, and Bamdev Mishra. Riemannian stochastic variance reduced gradient algorithm with retraction and vector transport. *SIAM Journal on Optimization*, 29(2):1444–1472, 2019. doi: 10.1137/17M1116787. URL *https://doi.org/10.1137/17M1116787*.

Weijie Su, Stephen P. Boyd, and Emmanuel J. Candès. A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights. *J. Mach. Learn. Res.*, 17:153:1–153:43, 2016. URL *http://jmlr.org/papers/v17/15-084.html*.

Ju Sun, Qing Qu, and John Wright. Complete dictionary recovery over the sphere II: recovery by Riemannian trust-region method. *IEEE Trans. Inf. Theory*, 63(2):885–914, 2017. doi: 10.1109/TIT.2016.2632149. URL *https://doi.org/10.1109/TIT.2016.2632149*.

Yue Sun, Nicolas Flammarion, and Maryam Fazel. Escaping from saddle points on Riemannian manifolds. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 7274–7284, 2019. URL *http://papers.nips.cc/paper/8948-escaping-from-saddle-points-on-riemannian-manifolds*.

Mingkui Tan, Ivor W. Tsang, Li Wang, Bart Vandereycken, and Sinno Jialin Pan. Riemannian pursuit for big matrix recovery. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 1539–1547, 2014. URL *http://proceedings.mlr.press/v32/tan14.html*.

Guo-ji Tang and Nan-Jing Huang. Rate of convergence for proximal point algorithms on Hadamard manifolds. *Oper. Res. Lett.*, 42(6-7):383–387, 2014. doi: 10.1016/j.orl.2014.06.009. URL *https://doi.org/10.1016/j.orl.2014.06.009*.

Nilesh Tripuraneni, Nicolas Flammarion, Francis Bach, and Michael I. Jordan. Averaging stochastic gradient descent on Riemannian manifolds. *CoRR*, abs/1802.09128, 2018. URL *http://arxiv.org/abs/1802.09128*.

Paul Tseng. On accelerated proximal gradient methods for convex-concave optimization. *submitted to SIAM Journal on Optimization*, 2(3), 2008. URL *https://www.mit.edu/~dimitrib/PTseng/papers/apgm.pdf*.

Bart Vandereycken. Low-rank matrix completion by Riemannian optimization. *SIAM Journal on Optimization*, 23(2):1214–1236, 2013. doi: 10.1137/110845768. URL *https://doi.org/10.1137/110845768*.

Rolf Walter. On the metric projection onto convex sets in Riemannian spaces. *Archiv der Mathematik*, 25(1):91–98, 1974. URL *http://www.afhalifax.ca/magazine/wp-content/sciences/EmpilementDeDisques/delaunaycourbe/YAMABEflow/geodesic%20convexity/KEYpaper.pdf*.

Di Wang, Satish Rao, and Michael W. Mahoney. Unified acceleration method for packing and covering problems via diameter reduction. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 50:1–50:13, 2016a. doi: 10.4230/LIPIcs.ICALP.2016.50. URL *https://doi.org/10.4230/LIPIcs.ICALP.2016.50*.

Jinhua Wang, Chong Li, Genaro López-Acedo, and Jen-Chih Yao. Convergence analysis of inexact proximal point algorithms on Hadamard manifolds. *J. Glob. Optim.*, 61(3):553–573, 2015. doi: 10.1007/s10898-014-0182-2. URL *https://doi.org/10.1007/s10898-014-0182-2*.

Jinhua Wang, Chong Li, Genaro López-Acedo, and Jen-Chih Yao. Proximal point algorithms on Hadamard manifolds: Linear convergence and finite termination. *SIAM J. Optim.*, 26(4):2696–2729, 2016b. doi: 10.1137/15M1051257. URL *https://doi.org/10.1137/15M1051257*.

Melanie Weber and Suvrit Sra. Frank-Wolfe methods for geodesically convex optimization with application to the matrix geometric mean. *CoRR*, abs/1710.10770, 2017. URL *http://arxiv.org/abs/1710.10770*.

Melanie Weber and Suvrit Sra. Nonconvex stochastic optimization on manifolds via Riemannian Frank-Wolfe methods. *CoRR*, abs/1910.04194, 2019. URL *http://arxiv.org/abs/1910.04194*.

Ke Wei, Jian-Feng Cai, Tony F Chan, and Shingyu Leung. Guarantees of Riemannian optimization for low rank matrix completion. *arXiv preprint arXiv:1603.06610*, 2016. URL *https://arxiv.org/abs/1603.06610*.

Andre Wibisono, Ashia C. Wilson, and Michael I. Jordan. A variational perspective on accelerated methods in optimization. *CoRR*, abs/1603.04245, 2016. URL *http://arxiv.org/abs/1603.04245*.

Ami Wiesel. Geodesic convexity and covariance estimation. *IEEE Trans. Signal Process.*, 60(12): 6182–6189, 2012. doi: 10.1109/TSP.2012.2218241. URL *https://doi.org/10.1109/TSP.2012.2218241*.

Hongyi Zhang and Suvrit Sra. First-order methods for geodesically convex optimization. In *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*, pages 1617–1638, 2016. URL *http://proceedings.mlr.press/v49/zhang16b.html*.

Hongyi Zhang and Suvrit Sra. An estimate sequence for geodesically convex optimization. In Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet, editors, *Conference On Learning Theory, COLT 2018, Stockholm, Sweden, 6-9 July 2018*, volume 75 of *Proceedings of Machine Learning Research*, pages 1703–1723. PMLR, 2018. URL *http://proceedings.mlr.press/v75/zhang18a.html*.

Hongyi Zhang, Sashank J. Reddi, and Suvrit Sra. Riemannian SVRG: fast stochastic optimization on Riemannian manifolds. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4592–4600, 2016. URL *http://papers.nips.cc/paper/6515-riemannian-svrg-fast-stochastic-optimization-on-riemannian-manifolds*.

Peiyuan Zhang, Jingzhao Zhang, and Suvrit Sra. Minimax in geodesic metric spaces: Sion's theorem and algorithms. *CoRR*, abs/2202.06950, 2022. URL *https://arxiv.org/abs/2202.06950*.

Pan Zhou, Xiao-Tong Yuan, and Jiashi Feng. Faster first-order methods for stochastic non-convex optimization on Riemannian manifolds. In *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, pages 138–147, 2019. URL *http://proceedings.mlr.press/v89/zhou19a.html*.

Appendix A. Convergence of Riemacon (Algorithm 1)

We start the analysis by noting a property that our parameters satisfy.

Lemma 7 *For the parameter choices of a_k and A_{k-1} in Algorithm 1 we have, for all $k \geq 1$:*

$$\frac{8\lambda}{9}(\xi A_{k-1} + a_k) \geq a_k^2 \geq \frac{3\lambda}{4}(\xi A_{k-1} + \xi a_k).$$

Proof It is a simple computation to check that a_k and A_{k-1} satisfy such inequality. The inequalities are equivalent to the following, which trivially holds:

$$\begin{aligned}\frac{8}{9}((k^2 - k + 64k\xi - 64\xi + 1000\xi^2) + (2k + 64\xi)) &\geq \frac{4}{5}(k^2 + 64k\xi + 1024\xi^2)\\ &\geq \frac{3}{4}((k^2 - k + 64k\xi - 64\xi + 1000\xi^2) + (2k\xi + 64\xi^2)).\end{aligned}$$

■

We now prove Proposition 3, which will allow us to use ψ_k as a Lyapunov function to show the final convergence rates. The proof will use Lemma 8 and Lemma 9, that we state and prove afterwards.

Proof of Proposition 3. The main ideas of the design of Algorithm 1 and of the proof of this proposition are the following. We run a type of mirror descent that estimates with a quadratic

Inequality $(1 - \Delta_k)\psi_k \leq \psi_{k-1}$ is equivalent to

$$\begin{aligned}&(1 - \Delta_k)\left(A_k(f(y_k) - f(\bar{x}^*)) + \frac{1}{2}\|z_k^{y_k} - \bar{x}^*\|_{y_k}^2 + \frac{\xi - 1}{2}\|y_k - z_k^{y_k}\|_{y_k}^2\right)\\ &\leq A_{k-1}(f(y_{k-1}) - f(\bar{x}^*)) + \left(\frac{1}{2}\|z_{k-1}^{y_{k-1}} - \bar{x}^*\|_{y_{k-1}}^2 + \frac{\xi - 1}{2}\|y_{k-1} - z_{k-1}^{y_{k-1}}\|_{y_{k-1}}^2\right)\end{aligned}$$

which, by bounding $(1 - \Delta_k)(f(y_k) - f(\bar{x}^*)) \leq f(y_k) - f(\bar{x}^*)$ and reorganizing, is implied by the following:

$$\begin{aligned}A_{k-1}(f(y_k) - f(y_{k-1})) + \frac{a_k}{\xi}(f(y_k) - f(\bar{x}^*)) &\leq \frac{1}{2}\|z_{k-1}^{y_{k-1}} - \bar{x}^*\|_{y_{k-1}}^2 - \frac{1 - \Delta_k}{2}\|z_k^{y_k} - \bar{x}^*\|_{y_k}^2\\ &+ \frac{\xi - 1}{2}\left(\|y_{k-1} - z_{k-1}^{y_{k-1}}\|_{y_{k-1}}^2 - (1 - \Delta_k)\|y_k - z_k^{y_k}\|_{y_k}^2\right).\end{aligned}$$

We have that due to the projection in Line 12, then x_k is not very far from any $p \in \mathcal{X}$:

$$d(x_k, p) \leq \|x_k - y_{k-1}\|_{y_{k-1}} + d(y_{k-1}, p) \overset{①}{<} \|\bar{z}_{k-1}^{y_{k-1}} - y_{k-1}\|_{y_{k-1}} + D \overset{②}{\leq} 2D, \tag{1}$$

where ① holds by the definition of x_k and the fact $y_{k-1}, p \in \mathcal{X}$, and ② is due to the projection defining $\bar{z}_{k-1}^{y_{k-1}}$. Now we use the first part of Lemma 8 with both $x \leftarrow y_{k-1}$ and $x \leftarrow \bar{x}^*$ and we bound the resulting errors $\varepsilon_k(\cdot)$ by using the second part of Lemma 8. We also use Lemma 9, so it is enough to prove the following:

$$\begin{aligned}&A_{k-1}\langle v_k^x, x_k - y_{k-1}\rangle + (a_k/\xi)\langle v_k^x, x_k - z_{k-1}^{x_k} + z_{k-1}^{x_k} - \bar{x}^*\rangle - \frac{4\lambda}{9}(A_{k-1} + a_k/\xi)\|v_k^x\|^2\\ &\leq \frac{1}{2}\|z_{k-1}^{x_k} - \bar{x}^*\|_{x_k}^2 - \frac{1}{2}\|z_k^{x_k} - \bar{x}^*\|_{x_k}^2 + \frac{\xi - 1}{2}\left(\|x_k - z_{k-1}^{x_k}\|_{x_k}^2 - \|x_k - z_k^{x_k}\|_{x_k}^2\right),\end{aligned}$$

where $z_{k-1}^{x_k} \stackrel{\text{def}}{=} \operatorname{Log}_{x_k}(\operatorname{Exp}_{y_{k-1}}(\bar{z}_{k-1}^{y_{k-1}}))$ and so $z_k^{x_k} = z_{k-1}^{x_k} - \eta_k v_k^x$. Note that thanks to Lemma 9 now we have the potentials on the right hand side as expressions in the tangent space of x_k. Also, note that we have canceled some potentials proportional to Δ_k coming from the bound on the error $\varepsilon_k(\cdot)$. Now we use that by definition of x_k we have, for all $v \in T_{x_k}\mathcal{M}$, $A_{k-1}\langle v, x_k - y_{k-1}\rangle = -a_k\langle v, x_k - z_{k-1}^{x_k}\rangle$, so we use this fact for $v = v_k^x$ and use the following identity, that holds by the definion of $z_k^{x_k} \stackrel{\text{def}}{=} z_{k-1}^{x_k} - \eta_k v_k^x$:

$$\frac{a_k/\xi}{\eta_k}\langle \eta_k v_k^x, z_{k-1}^{x_k} - \bar{x}^*\rangle = \frac{a_k/\xi}{2\eta_k}\left(\eta_k^2\|v_k^x\|_{x_k}^2 + \|z_{k-1}^{x_k} - \bar{x}^*\|_{x_k}^2 - \|z_k^{x_k} - \bar{x}^*\|_{x_k}^2\right).$$

so that, after canceling terms, it is enough to prove:

$$\begin{aligned} a_k(1-1/\xi)\langle -v_k^x, x_k - z_{k-1}^{x_k}\rangle - \frac{a_k(1-1/\xi)}{2\eta_k}\eta_k^2\|v_k^x\|^2 \\ + \|v_k^x\|^2\left(-\frac{4}{9}(A_{k-1}\lambda + a_k\lambda/\xi) + \frac{a_k\eta_k}{2}\right) \\ \leq \frac{\xi-1}{2}\left(\|x_k - z_{k-1}^{x_k}\|_{x_k}^2 - \|x_k - z_k^{x_k}\|_{x_k}^2\right), \end{aligned} \tag{2}$$

Now we show that in the previous inequality (2), the first line cancels with the last line. Note that $(a_k(1-1/\xi))/\eta_k = (1-1/\xi)/(1/\xi) = \xi - 1$. Thus, by using again the definition of $z_k^{x_k}$, we have:

$$\frac{a_k(1-1/\xi)}{\eta_k}\langle -\eta_k v_k^x, x_k - z_{k-1}^{x_k}\rangle = \frac{a_k(1-1/\xi)}{2\eta_k}\left(\eta_k^2\|v_k^x\|_{x_k}^2 + \|x_k - z_{k-1}^{x_k}\|_{x_k}^2 - \|x_k - z_k^{x_k}\|_{x_k}^2\right).$$

Finally, it only remains to prove:

$$\frac{\|v_k^x\|^2}{2\xi}\cdot\left(-\frac{8}{9}(\xi A_{k-1}\lambda + a_k\lambda) + a_k^2\right) \leq 0, \tag{3}$$

which holds by Lemma 7. ∎

We now show the two auxiliary lemmas that we used in the previous proof.

Lemma 8 *Let $h_k(x) \stackrel{\text{def}}{=} f(x) + \frac{1}{2\lambda}d(x_k, x)^2$ be the strongly g-convex function used at step k, and let $y_k^* = \arg\min_{y\in\mathcal{X}} h_k(y)$. Then, for $y_k \in \mathcal{X}$, if we let $v_k^x \stackrel{\text{def}}{=} -\operatorname{Log}_{x_k}(y_k)/\lambda$, then the following holds, for all $x \in \mathcal{X}$:*

$$f(x) \geq f(y_k) + \langle v_k^x, x - x_k\rangle_{x_k} + \frac{\lambda}{2}\|v_k^x\|^2 - \varepsilon_k(x)$$

where $\varepsilon_k(x) \stackrel{\text{def}}{=} -\frac{1}{\lambda}\langle y_k - y_k^*, x - x_k\rangle_{x_k} + (h_k(y_k) - h_k(y_k^*))$. Moreover, if y_k satisfies

$$h_k(y_k) - h_k(y_k^*) \leq \frac{\Delta_k d(x_k, y_k^*)^2}{78\lambda},$$

then we have

$$\begin{aligned} -\frac{\lambda}{2}\|v_k^x\|^2(A_{k-1} + a_k/\xi) + a_k\varepsilon_k(\bar{x}^*)/\xi + A_{k-1}\varepsilon_k(y_{k-1}) \\ \leq -\frac{4\lambda\|v_k^x\|^2}{9}(A_{k-1} + a_k/\xi) + \frac{\Delta_k}{2}\left(\|\bar{x}^* - z_{k-1}^{x_k}\|_{x_k}^2 + (\xi - 1)\|x_k - z_{k-1}^{x_k}\|_{x_k}^2\right). \end{aligned}$$

Proof The function h_k is $\frac{1}{\lambda}$-strongly g-convex because by Fact 2 the function $\frac{1}{2}d(x_k, x)^2$ is 1-strongly g-convex in a Hadamard manifold. By the first-order optimality condition of h_k at y_k^* we have that $\tilde{v}_k^y \stackrel{\mathrm{def}}{=} \lambda^{-1}\mathrm{Log}_{y_k^*}(x_k) \in \partial(f + I_{\mathcal{X}})(y_k^*)$ is a subgradient of $f + I_{\mathcal{X}}$ at y_k^*. Thus, we have, for all $x \in \mathcal{X}$ and for $\tilde{v}_k^x \stackrel{\mathrm{def}}{=} \Gamma_{y_k^*}^{x_k}(\tilde{v}_k^y)$:

$$
\begin{aligned}
f(x) &\overset{①}{\geq} f(y_k^*) + \langle \tilde{v}_k^y, x - y_k^* \rangle_{y_k^*} \\
&\overset{②}{\geq} f(y_k^*) + \langle \tilde{v}_k^x, x - x_k \rangle_{x_k} + \lambda \|\tilde{v}_k^x\|^2 \\
&\overset{③}{=} f(y_k) + \langle v_k^x, x - x_k \rangle_{x_k} + \frac{\lambda}{2}\|v_k^x\|^2 + \frac{\lambda}{2}\|\tilde{v}_k^x\|^2 \\
&\quad + \langle \tilde{v}_k^x - v_k^x, x - x_k \rangle_{x_k} + \left((f(y_k^*) + \frac{\lambda}{2}\|\tilde{v}_k^x\|^2) - (f(y_k) + \frac{\lambda}{2}\|v_k^x\|^2) \right) \\
&\overset{④}{\geq} f(y_k) + \langle v_k^x, x - x_k \rangle_{x_k} + \frac{\lambda}{2}\|v_k^x\|^2 + \frac{1}{\lambda}\langle y_k - y_k^*, x - x_k \rangle_{x_k} - (h_k(y_k) - h_k(y_k^*))
\end{aligned}
$$

where ① holds because $\tilde{v}_k^y \in \partial(f + I_{\mathcal{X}})(y_k^*)$ and $x, y_k^* \in \mathcal{X}$. In ②, we used the first part of Lemma 17 along with $\bar{\delta} = 1$. We just added and subtracted some terms in ③, and in ④, we dropped $\frac{\lambda}{2}\|\tilde{v}_k^x\|^2$, and we used the definitions of h_k, $\tilde{v}_k^x$, and $v_k^x = -\mathrm{Log}_{x_k}(y_k)/\lambda$.

Now we proceed to prove the second part. The following holds:

$$
\begin{aligned}
&-\frac{a_k}{\lambda\xi}\langle y_k - y_k^*, \bar{x}^* - x_k \rangle_{x_k} - A_{k-1}\frac{1}{\lambda}\langle y_k - y_k^*, y_{k-1} - x_k \rangle_{x_k} \\
&\overset{①}{\leq} \frac{1}{\lambda}\|y_k - y_k^*\|_{x_k} \cdot \|\frac{a_k}{\xi}\bar{x}^* + A_{k-1}y_{k-1} - (\frac{a_k}{\xi} + A_{k-1})x_k\|_{x_k} \\
&\overset{②}{\leq} \frac{1}{\lambda}d(y_k, y_k^*) \cdot \frac{a_k}{\xi}\|\bar{x}^* - z_{k-1}^{x_k} + (\xi - 1)(x_k - z_{k-1}^{x_k})\|_{x_k} \\
&\overset{③}{\leq} \frac{1}{\lambda}\sqrt{2\lambda(h_k(y_k) - h_k(y_k^*))} \cdot \frac{a_k}{\xi}\sqrt{\xi}\sqrt{\|\bar{x}^* - z_{k-1}^{x_k}\|_{x_k}^2 + (\xi - 1)\|(x_k - z_{k-1}^{x_k})\|_{x_k}^2} \\
&= \sqrt{\frac{2a_k^2(h_k(y_k) - h_k(y_k^*))}{\Delta_k \lambda\xi}} \cdot \sqrt{\Delta_k}\sqrt{\|\bar{x}^* - z_{k-1}^{x_k}\|_{x_k}^2 + (\xi - 1)\|(x_k - z_{k-1}^{x_k})\|_{x_k}^2} \\
&\overset{④}{\leq} \frac{a_k^2(h_k(y_k) - h_k(y_k^*))}{\Delta_k \lambda\xi} + \frac{\Delta_k}{2}(\|\bar{x}^* - z_{k-1}^{x_k}\|_{x_k}^2 + (\xi - 1)\|(x_k - z_{k-1}^{x_k})\|_{x_k}^2),
\end{aligned}
\tag{4}
$$

where ① groups some terms and uses Cauchy-Schwartz. In inequality ②, for the first term we bounded the distance between y_k^* and y_k estimated from $T_{x_k}\mathcal{M}$ by the actual distance, which is a property that holds in Hadamard manifolds and it holds by the first part of Corollary 14 with $\bar{\delta} = 1$, $p \leftarrow y_k^*$, $y \leftarrow y_k$, $x \leftarrow x_k$, $z^y \leftarrow 0$. The second term is substituted by a term of equal value by using Euclidean trigonometry in $T_{x_k}\mathcal{M}$, as in the following. Let $w \stackrel{\mathrm{def}}{=} \frac{1}{a_k/\xi + A_{k-1}}(\frac{a_k}{\xi}\mathrm{Log}_{x_k}(\bar{x}^*) + A_{k-1}\mathrm{Log}_{x_k}(y_{k-1}))$ and let $u \in T_{x_k}$ be the point in the line containing $\mathrm{Log}_{x_k}(y_{k-1})$ and $0 = \mathrm{Log}_{x_k}(x_k) \in T_{x_k}$ such that the triangle with vertices 0, $\mathrm{Log}_{x_k}(y_{k-1})$ and w and the triangle with vertices u, $\mathrm{Log}_{x_k}(y_{k-1})$ and $\mathrm{Log}_{x_k}(\bar{x}^*)$ are similar triangles, and so

$$\frac{\|\mathrm{Log}_{x_k}(\bar{x}^*) - u\|}{\|w - \mathrm{Log}_{x_k}(x_k)\|} \overset{⑤}{=} \frac{\|\mathrm{Log}_{x_k}(\bar{x}^*) - \mathrm{Log}_{x_k}(y_{k-1})\|}{\|w - \mathrm{Log}_{x_k}(y_{k-1})\|} \overset{⑥}{=} \frac{A_{k-1} + a_k/\xi}{a_k/\xi}. \tag{5}$$

We used the triangle similarity in ⑤ and in ⑥ we used the definition of w as a convex combination of $\mathrm{Log}_{x_k}(\bar{x}^*)$ and $\mathrm{Log}_{x_k}(y_{k-1})$. It is enough to show $u = \xi z_{k-1}^{x_k}$ as in such a case what we applied in ② is equivalent to the equality (5) above. By the definition of x_k, we have ⑦ below and by triangle similarity we have ⑧ below:

$$z_{k-1}^{x_k} \overset{⑦}{=} -\frac{A_{k-1}}{a_k}\mathrm{Log}_{x_k}(y_{k-1}) \overset{⑧}{=} \frac{A_{k-1}}{a_k} \cdot \frac{a_k/\xi}{A_{k-1}} u = \frac{1}{\xi} u,$$

as desired. In the next inequality ③, we used that by $(1/\lambda)$-strong g-convexity of h_k and by optimality of y_k^*, we have $\frac{1}{2\lambda} d(\cdot, y_k^*)^2 \leq h_k(\cdot) - h_k(y_k^*)$. For the second term, we used that for vectors $b, c \in \mathbb{R}^n$ and $\omega \in \mathbb{R}_{\geq 0}$, we have, by Young's inequality, $\|b + wc\| = \sqrt{\|b\|^2 + \omega^2\|c\|^2 + 2\langle\sqrt{\omega}b, \sqrt{\omega}c\rangle} \leq \sqrt{(1+\omega)(\|b\|^2 + \omega\|c\|^2)}$. In ④ we used Young's inequality.

Before we conclude, we note that

$$d(x_k, y_k^*) \leq \sqrt{2} d(x_k, y_k), \tag{6}$$

which is implied by the following, where we use the same as in ③ above, the assumption on y_k and $\Delta_k \leq 1$:

$$\begin{aligned} d(x_k, y_k^*) &\leq d(x_k, y_k) + d(y_k, y_k^*) \leq d(x_k, y_k) + \sqrt{2\lambda(h_k(y_k) - h_k(y_k^*))} \\ &\leq d(x_k, y_k) + \sqrt{\Delta_k/34} \cdot d(x_k, y_k^*) \leq d(x_k, y_k) + d(x_k, y_k^*)/4. \end{aligned}$$

Finally, we can make use of (4) and (6) to obtain the claim in the second part of the lemma:

$$\begin{aligned} &-\frac{\lambda}{2}\|v_k^x\|^2(A_{k-1} + a_k/\xi) + a_k\varepsilon_k(\bar{x}^*)/\xi + A_{k-1}\varepsilon_k(y_{k-1}) - \frac{\Delta_k}{2}\|\bar{x}^* - z_{k-1}^{x_k}\|_{x_k}^2 \\ &\quad - \Delta_k \frac{\xi - 1}{2}\|(x_k - z_{k-1}^{x_k})\|_{x_k}^2 \\ &\quad \leq -\frac{\lambda}{2}\|v_k^x\|^2(A_{k-1} + a_k/\xi) + \left(A_{k-1} + a_k/\xi + \frac{a_k^2}{\Delta_k \lambda \xi}\right)(h_k(y_k) - h_k(y_k^*)) \\ &\quad \overset{①}{\leq} -\frac{\lambda}{2}\|v_k^x\|^2(A_{k-1} + a_k/\xi) + (A_{k-1} + a_k/\xi)\left(1 + \frac{a_k^2}{(\xi A_{k-1} + a_k)\lambda}\right)\frac{d(x_k, y_k)^2}{34\lambda} \\ &\quad \overset{②}{\leq} -\frac{\lambda}{2}\|v_k^x\|^2(A_{k-1} + a_k/\xi) + \frac{d(x_k, y_k)^2}{18\lambda}(A_{k-1} + a_k/\xi) \\ &\quad \overset{③}{=} -\frac{4\lambda\|v_k^x\|^2}{9}(A_{k-1} + a_k/\xi), \end{aligned}$$

where ① holds by the assumption on y_k, $\Delta_k \leq 1$, and (6). Inequality ② uses the upper bound on a_k^2 in Lemma 7, and ③ uses the definition $v_k^x \overset{\mathrm{def}}{=} -\mathrm{Log}_{x_k}(y_k)/\lambda$.

■

The following lemma allows to *move* the regularized lower bounds on the objective without incurring extra geometric penalties.

Lemma 9 (Translating Potentials with no Geometric Penalty) *Using the variables in Algorithm 1, for any* $\Delta_k \in [0,1)$*, we have*

$$\begin{aligned}\|z_{k-1}^{x_k} - \bar{x}^*\|_{x_k}^2 - (1-\Delta_k)\|z_k^{x_k} - \bar{x}^*\|_{x_k}^2 + (\xi-1)\left(\|x_k - z_{k-1}^{x_k}\|_{x_k}^2 - (1-\Delta_k)\|x_k - z_k^{x_k}\|_{x_k}^2\right) \\ \leq \|z_{k-1}^{y_{k-1}} - \bar{x}^*\|_{y_{k-1}}^2 - (1-\Delta_k)\|z_k^{y_k} - \bar{x}^*\|_{y_k}^2 \\ + (\xi-1)\left(\|y_{k-1} - z_{k-1}^{y_{k-1}}\|_{y_{k-1}}^2 - (1-\Delta_k)\|y_k - z_k^{y_k}\|_{y_k}^2\right).\end{aligned}$$

Proof Recall that we defined $z_{k-1}^{x_k} \stackrel{\text{def}}{=} \operatorname{Log}_{x_k}(\operatorname{Exp}_{y_{k-1}}(\bar{z}_{k-1}^{y_{k-1}}))$ in the proof of Proposition 3. Firstly, by the projection step in Line 12, we have

$$\|z_{k-1}^{y_{k-1}} - \bar{x}^*\|_{y_k}^2 \geq \|\bar{z}_{k-1}^{y_{k-1}} - \bar{x}^*\|_{y_k}^2 \qquad \text{and} \qquad (\xi-1)\|z_{k-1}^{y_{k-1}}\|_{y_k}^2 \geq (\xi-1)\|\bar{z}_{k-1}^{y_{k-1}}\|_{y_k}^2 \tag{7}$$

since the operation is a simple Euclidean projection onto the closed ball $\bar{B}(0,D)$ in $T_{y_k}\mathcal{M}$. By the second part of Corollary 14, $y = x_k$ and $x = y_{k-1}$ and by (1), we have ① below

$$\begin{aligned}\|\bar{z}_{k-1}^{y_{k-1}} - \bar{x}^*\|_{y_{k-1}}^2 + (\xi-1)\|\bar{z}_{k-1}^{y_{k-1}}\|_{y_{k-1}}^2 &\overset{①}{\geq} \|z_{k-1}^{x_k} - \bar{x}^*\|_{x_k}^2 + (\zeta_{2D}-1)\|z_{k-1}^{x_k}\|_{x_k}^2 + (\xi-\zeta_{2D})\|\bar{z}_{k-1}^{y_{k-1}}\|_{y_{k-1}}^2 \\ &\overset{②}{\geq} \|z_{k-1}^{x_k} - \bar{x}^*\|_{x_k}^2 + (\xi-1)\|z_{k-1}^{x_k}\|_{x_k}^2 + (\xi-\zeta_{2D})\left(\left(\frac{A_{k-1}+a_k}{A_{k-1}}\right)^2 - 1\right)\|z_{k-1}^{x_k}\|_{x_k}^2 \\ &\overset{③}{\geq} \|z_{k-1}^{x_k} - \bar{x}^*\|_{x_k}^2 + (\xi-1)\|z_{k-1}^{x_k}\|_{x_k}^2 + \frac{3(\xi-1)}{2}\left(\frac{1}{1-\tau_k} - 1\right)\|z_{k-1}^{x_k}\|_{x_k}^2,\end{aligned} \tag{8}$$

and ② uses the definition of x_k. In ③, we used the definition of $\xi = 4\zeta_{2D} - 3$ that implies $\xi - \zeta_{2D} \geq \frac{3}{4}(\xi-1)$ and for $\tau_k \stackrel{\text{def}}{=} a_k/(a_k + A_{k-1})$ we have that $(1+\frac{a_k}{A_{k-1}})^2 - 1 \geq \frac{2a_k}{A_{k-1}} = 2(\frac{1}{1-\tau_k} - 1)$. Now, we use the second part of Lemma 13 with $y = y_k$, $x = x_k$ $z^x = -\eta_k v_k^x$, $a^x = z_{k-1}^{x_k}$, so that $z^x + a^x = z_k^{x_k}$ and $z^y + a^y = z_k^{y_k}$ and

$$r = \frac{\|\operatorname{Log}_{x_k}(y_k)\|}{\|z^x\|} = \frac{\lambda\|v_k^x\|}{\eta_k\|v_k^x\|} = \frac{\xi\lambda}{a_k} = \frac{5\xi}{2k+64\xi} < 5/6 < 1. \tag{9}$$

Note that by the choice of parameters and the fact that $r < 1$, the assumptions in Lemma 13 are satisfied. Thus, the following holds

$$\|z_k^{x_k} - \bar{x}^*\|_{x_k}^2 + (\xi-1)\|z_k^{x_k}\|_{x_k}^2 + \frac{\xi-1}{2}\left(\frac{r}{1-r}\right)\|z_{k-1}^{x_k}\|^2 \geq \|z_k^{y_k} - \bar{x}^*\|_{y_k}^2 + (\xi-1)\|z_k^{y_k}\|_{y_k}^2. \tag{10}$$

Hence, combining (7), (8) and (10) we obtain that it is enough to prove

$$-(1-\Delta_k)\left(\frac{r}{1-r}\right)+3\left(\frac{1}{1-\tau_k}-1\right)\geq 0,$$

The proof will be finished if we prove the result for $\Delta_k = 0$. If we use this last inequality, and the fact that for $r \leq 5/6$, we have $\frac{r}{1-r} \leq 3\left(\frac{1}{1-3r/4}-1\right)$, we deduce that it suffices to show $\tau_k \geq \frac{3}{4}r$ to conclude

$$\frac{r}{1-r}\leq 3\left(\frac{1}{1-3r/4}-1\right)\leq 3\left(\frac{1}{1-\tau_k}-1\right).$$

Such inequality, namely $\tau_k \geq \frac{3}{4}r$, is equivalent to $\frac{a_k^2}{\lambda} \geq \frac{3\xi}{4}(a_k + A_{k-1})$ and it holds by Lemma 7.
■

Finally, we use Proposition 3 to show the final convergence rates. The proof will also use Lemma 10 that is stated and proved after the proof of the theorem.

Proof of Theorem 4. Given the inequality $(1-\Delta_k)\psi_k \leq \psi_{k-1}$, proven in Proposition 3, we can use ψ_k as a Lyapunov function in order to prove convergence rates of Algorithm 1. It follows straightforwardly by definition of ψ_k, in the following way

$$\begin{aligned}
f(y_k)-f(\bar{x}^*) \leq \frac{\psi_k}{A_k} \leq \prod_{i=1}^{k}(1-\Delta_i)^{-1}\frac{\psi_0}{A_k} &\overset{①}{\leq} \frac{2\psi_0}{A_k} \overset{②}{=} O\left(\frac{\bar{R}^2}{\lambda}\left(\frac{A_0}{A_k}+\frac{\lambda}{A_k}\right)\right) \\
&= O\left(\frac{\bar{R}^2}{\lambda}\left(\frac{\xi}{\frac{k^2+\xi k}{\xi}+\xi}+\frac{1}{\frac{k^2+\xi k}{\xi}+\xi}\right)\right) \\
&= O\left(\frac{\bar{R}^2}{\lambda}\left(\frac{\xi^2}{k^2+\xi k+\xi^2}\right)\right) \overset{③}{=} O\left(\frac{\bar{R}^2}{\lambda k^2}\cdot\bar{\zeta}^2\right).
\end{aligned}$$

In ①, we used $\prod_{i=1}^{k}(1-\Delta_k) = \prod_{i=1}^{k}\frac{i(i+2)}{(i+1)^2} = \frac{k+2}{2(k+1)} \geq \frac{1}{2}$. Now for ②, we note the following, which is analogous to applying smoothness of the Riemannian Moreau envelope to show that an inexact prox step serves as a warm start:

$$\begin{aligned}
f(y_0)-f(\bar{x}^*) \leq f(y_0)+\frac{1}{2\lambda}d(x_0,y_0)^2-f(\bar{x}^*) &\overset{④}{\leq} \min_{y\in\mathcal{X}}\{f(y)+\frac{1}{2\lambda}d(x_0,y)^2\}+\sigma_0-f(\bar{x}^*) \\
&\overset{⑤}{\leq} \frac{1}{2\lambda}d(x_0,\bar{x}^*)^2+\frac{1}{78\lambda}d(x_0,y_0^*)^2 \overset{⑥}{=} O(\bar{R}^2/\lambda).
\end{aligned}$$

We used σ_0-optimality of y_0 in ④. In ⑤ we substituted the value of σ_0 and we set $y \leftarrow \bar{x}^*$ in the minimum. We used Lemma 10 for ⑥. In ②, we also used $\frac{\xi-1}{2}\|z_0^{y_0}\|_{y_0}^2 = 0$ and

$$\|z_0^{y_0}-\bar{x}^*\|_{y_0} = d(y_0,x^*) \leq d(y_0,y_0^*)+d(y_0^*,\bar{x}^*) \overset{⑦}{\leq} \sqrt{2\lambda(h_0(y_0)-h_0(y_0^*))}+\bar{R} \overset{⑧}{=} O(\bar{R}),$$

where ⑦ holds by 1-strong convexity of λh_0 and Lemma 10, while in ⑧ we used the σ_0-optimality of y_0 and Lemma 10.

In ③, we used $\xi = O(\bar{\zeta})$ and we dropped some terms in the denominator. This means that the number of iterations is $O(\bar{\zeta}\sqrt{\frac{\bar{R}^2}{\lambda\varepsilon}})$ if we want the right hand side to be bounded by ε.

The algorithm and analysis for strongly g-convex and smooth functions follows directly by applying the reduction in (Martínez-Rubio, 2020, Theorem 7) to Algorithm 1. We denote this algorithm by RiemaconSC$(\mathcal{X}, x_0, f, \lambda, \varepsilon)$, where $\mathcal{X}$ is the feasible set, x_0 is the initial point, f is the function to optimize, λ is the implicit gradient descent learning rate, and ε is an optional parameter specifying the desired accuracy. Although the statement of the reduction in this paper assumes an L-smooth function $f : \mathcal{M} \to \mathbb{R}$ to be optimized has a global minimizer in an unconstrained problem, the same proof of this theorem works if we have a μ-strongly g-convex function f defined over an open set containing a closed g-convex set $\mathcal{X}$ and a minimizer $\bar{x}^*$ of this function restricted to $\mathcal{X}$. The algorithm runs the algorithm for g-convex smooth minimization for $\mathrm{Time}_{\mathrm{ns}}(\mu, R)$, where this is defined as the number of iterations needed by the non-strongly g-convex algorithm to reach accuracy $\mu R^2/8$ if the initial distance is upper bounded by R. In such a case it guaranteed that the distance to the minimizer is reduced by half, and we restart the algorithm and run it again with the initial distance parameter equal to $R/2$, and so on. This happens $O(\log(\mu\bar{R}^2\varepsilon))$ times if we want to achieve accuracy ε from an initial distance $\bar{R}$. Thus, the total complexity in number of iterations can be bounded by $O(\mathrm{Time}_{\mathrm{ns}}(\mu, \bar{R})\log(\mu\bar{R}^2/\varepsilon))$, since all initial distances are $\leq \bar{R}$. In our case, since we optimize over the set $\mathcal{X}$ with diameter D, so it is $\mathrm{Time}_{\mathrm{ns}}(\mu, R) = O(\bar{\zeta}(\lambda\mu)^{-1/2} + 1)$, and the total number of iterations is $O((\bar{\zeta}(\lambda\mu)^{-1/2} + 1)\log(\mu\bar{R}^2/\varepsilon))$. We note that the reverse reduction in (Martínez-Rubio, 2020) yields extra geometric penalties but this one does not. ■

We now state and prove the lemma we used for Theorem 4.

Lemma 10 *For all $k \geq 0$, we have $d(x_k, y_k^*) \leq d(x_k, \bar{x}^*)$ and $d(y_k^*, \bar{x}^*) \leq d(x_k, \bar{x}^*)$.*

Proof Using $\bar{x}^* \in \arg\min_{x\in\mathcal{X}} f(x)$ and $y_k^* = \arg\min_{y\in\mathcal{X}} h_k(y)$, we have:

$$d(x_k, y_k^*)^2 - d(x_k, \bar{x}^*)^2 \leq 2\lambda f(y_k^*) + d(x_k, y_k^*)^2 - 2\lambda f(\bar{x}^*) - d(x_k, \bar{x}^*)^2 = 2\lambda(h_k(y_k^*) - h_k(\bar{x}^*)) \leq 0.$$

The second inequality is deduced from 1-strong convexity of λh_k, which holds by Fact 2 since we are in a Hadamard manifold, as well as the definition of h_k, and the fact $\bar{x}^* \in \arg\min_{x\in\mathcal{X}} f(x)$:

$$d(y_k^*, \bar{x}^*)^2 \leq 2\lambda(h_k(\bar{x}^*) - h_k(y_k^*)) \leq 2\lambda f(\bar{x}^*) + d(x_k, \bar{x}^*)^2 - 2\lambda f(y_k^*) \leq d(x_k, \bar{x}^*)^2.$$

■

Appendix B. Convergence of boosted Riemacon (Algorithm 2)

We start by showing that the iterates of Algorithm 2 stay reasonably bounded, which is crucial in order to bound geometric penalties.

Proposition 11 *The iterates x_k of Algorithm 2 satisfy $d(x_k, x^*) \leq 2R$.*

Proof We first show that the optimizer x_k^* in the ball $\mathcal{X}_k$ is no farther than the center of $\mathcal{X}_k$ to x^*, that is, $d(x_k^*, x^*) \leq d(x_{k-1}, x^*)$. We assume x^* is not in the ball because otherwise the property holds trivially. The geodesic segment joining x_k^* and x^* does not contain any other point of the ball, since otherwise by strong convexity we would have that the function value of one such point would be lower than $f(x_k^*)$. This fact implies that the angle between $\mathrm{Log}_{x_k^*}(x^*)$ and $\mathrm{Log}_{x_k^*}(x_{k-1})$ is obtuse, and so ① holds below and by using Corollary 15 we conclude $d(x_k^*, x^*) \leq d(x_{k-1}, x^*)$:

$$0 \overset{①}{\geq} 2\langle \mathrm{Log}_{x_k^*}(x^*), \mathrm{Log}_{x_k^*}(x_{k-1})\rangle \geq d(x_k^*, x^*)^2 + \delta \cdot d(x_k^*, x_{k-1})^2 - d(x_{k-1}, x^*)^2$$
$$\geq d(x_k^*, x^*)^2 - d(x_{k-1}, x^*)^2.$$

If instead of optimizing exactly in the ball we obtain a close approximation, the iterates will not get very far from x^*. Indeed, by μ-strong convexity, if x_k is an ε'-minimizer of f in $\mathcal{X}_k$, we have that $d(x_k^*, x_k) \leq \sqrt{\frac{2\varepsilon'}{\mu}} \leq \frac{R}{T}$, where we used the definition of $\varepsilon' = \min\{\frac{D\varepsilon}{8R}, \frac{\mu R^2}{2T^2}\}$ in the last inequality. Consequently, applying the non-expansiveness and this last inequality recursively, we obtain

$$d(x_T, x^*) \leq d(x_T^*, x^*) + d(x_T^*, x_T) \leq d(x_{T-1}, x^*) + \frac{R}{T} \leq \cdots \leq d(x_0, x^*) + R \leq 2R.$$

■

Before we prove Theorem 6, let's discuss about the initialization of D. As we explain in Appendix B.1, we can apply the subroutine in (Criscitiello and Boumal, 2021, Section 6) for any value of D that satisfies (notice D is twice the radius of the ball):

$$D \leq (46R|\kappa_{\min}|\zeta_D)^{-1}, \tag{11}$$

If $D = 2R$ satisfies the inequality, then the algorithm uses this value. If it is not satisfied, then for any value $D \geq 0$ that satisfies the inequality it must be $D < 2R$, so we assume that this inequality holds for the rest of the argument. Indeed, it is a consequence of the function $x^2\coth(x)$ being monotonously increasing for $x \geq 0$ and that given the definition of $\zeta_D = D\sqrt{|\kappa_{\min}|}\coth(D\sqrt{|\kappa_{\min}|})$, we have that inequality (11) is equivalent to $D^2|\kappa_{\min}|\coth(D\sqrt{|\kappa_{\min}|}) \leq (46R\sqrt{|\kappa_{\min}|})^{-1}$. In this case, the larger D is, the faster the algorithm runs. So one could solve the 1-dimensional problem $D = (46R|\kappa_{\min}|\zeta_D)^{-1}$ on D in order to obtain the best guarantee. On the other hand, we can provide the simple bound on this 1-dimensional problem $D = 1/(70R|\kappa_{\min}|)$ which would only lose a constant in the final convergence rates. We show now how this is indeed a bound. Let x be $D\sqrt{|\kappa_{\min}|}$, for some D satisfying inequality (11) and let S be the set of all such $x \geq 0$. Because we want $x^2\coth(x) \leq (46R\sqrt{|\kappa_{\min}|})^{-1}$ and the right hand side is upper bounded by $\leq 1/(23x)$, then by monotonicity it must be $S \subset [0, 1/4]$. It holds that for this interval the fourth derivative of $x^2\coth(x) \leq 0$, which along with its third order Taylor expansion yields ① below, so the points satisfying ③ below are in S and we can use $D = \frac{x}{\sqrt{|\kappa_{\min}|}} = \frac{1}{70R|\kappa_{\min}|} \leq \frac{3}{4\cdot 46R|\kappa_{\min}|}$ as our simple-to-compute bound:

$$x^2\coth(x) \overset{①}{\leq} x + \frac{x^3}{3} \overset{②}{\leq} \frac{4}{3}x \overset{③}{\leq} \frac{1}{46R\sqrt{|\kappa_{\min}|}}.$$

where in ② we used $x < 1$ for all $x \in S$. Now, we can proceed to prove the theorem.

Proof of Theorem 6. If $D = 2R$, which is the case in which the condition in Line 2 of Algorithm 2 is satisfied, then we just need to call Algorithm 1 once in the corresponding ball $\bar{B}(x_0, R)$ and we obtain rates $\widetilde{O}(\zeta^2\sqrt{\frac{L}{\mu}})$. So from now on we assume $D < 2R$. Let $T = \lceil \frac{4R}{D}\ln(\frac{LR^2}{\varepsilon})\rceil$ and let $\varepsilon' = \min\{\frac{D\varepsilon}{8R}, \frac{\mu R^2}{2T^2}\}$. Since every time we call Algorithm 1 we do it over a ball of diameter D, we still use the notation $\bar{\zeta} \stackrel{\text{def}}{=} \zeta_D$ to refer to the geometric constant associated to the sets $\mathcal{X}_k$, for every $k \geq 1$. Recall that we use $\zeta \stackrel{\text{def}}{=} \zeta_R = R\sqrt{|\kappa_{\min}|}\coth(R\sqrt{|\kappa_{\min}|}) \in [R\sqrt{|\kappa_{\min}|}, R\sqrt{|\kappa_{\min}|}+1]$.

By definition, it is $D \leq (46R|\kappa_{\min}|\bar{\zeta})^{-1}$. Using $2R > D$ and $\bar{\zeta} \in [D\sqrt{|\kappa_{\min}|}, D\sqrt{|\kappa_{\min}|}+1]$, we conclude $D \leq 1/\sqrt[3]{46}\sqrt{|\kappa_{\min}|} \leq 1/\sqrt{|\kappa_{\min}|}$ and $\bar{\zeta} \leq D\sqrt{|\kappa_{\min}|} + 1 \leq 2$. Since $\bar{\zeta} = O(1)$, the subroutine in Line 8 takes $\widetilde{O}(1)$ gradient oracle calls by the analysis in Appendix B.1 and thus, Line 10 of Algorithm 2 takes $\widetilde{O}(\sqrt{\frac{L}{\mu}}\log(\frac{1}{\varepsilon'}))$ gradient oracle calls to optimize in the ball $\mathcal{X}_k$ of diameter D, for any k. Recall that we denote the global optimizer of f by x^*. Define the g-convex combination

$$\tilde{x}_k = \mathrm{Exp}_{x_{k-1}}\left(\frac{D}{4R}\mathrm{Log}_{x_{k-1}}(x^*)\right) = \mathrm{Exp}_{x_{k-1}}\left((1-\frac{D}{4R})x_{k-1} + \frac{D}{4R}x^*\right).$$

Since $\mathcal{X}_k$ is a ball of radius $D/2$ and by Proposition 11, it is $d(x_k, x^*) \leq 2R$, we have $\tilde{x}_k \in \mathcal{X}_k$. Consequently, we have

$$f(x_k) \stackrel{①}{\leq} f(\tilde{x}_k) + \varepsilon' \stackrel{②}{\leq} (1-\frac{D}{4R})f(x_{k-1}) + \frac{D}{4R}f(x^*) + \varepsilon',$$

where ① is due to the guarantees of the optimization in the ball and the fact that $\tilde{x}_k \in \mathcal{X}_k$, ② holds due to g-convexity. Subtracting $f(x^*)$ in both sides and rearranging, we obtain

$$f(x_k) - f(x^*) \leq (1-\frac{D}{4R})(f(x_{k-1}) - f(x^*)) + \varepsilon'.$$

Applying this inequality recursively, we obtain

$$\begin{aligned} f(x_T) - f(x^*) &\leq (1-\frac{D}{4R})^T(f(x_0) - f(x^*)) + \varepsilon'\sum_{i=0}^{T-1}(1-\frac{D}{4R})^i \\ &\stackrel{①}{\leq} \exp(-\frac{DT}{4R})\frac{LR^2}{2} + \frac{4R}{D}\varepsilon' \\ &\leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon. \end{aligned}$$

Above, we used $1 - x \leq \exp(-x)$, we used smoothness to bound $f(x_0) - f(x^*) \leq \frac{Ld(x_0,x^*)^2}{2}$, we bounded $\sum_{i=0}^{T-1}(1-\frac{D}{4R})^i \leq \sum_{i=0}^{\infty}(1-\frac{D}{4R})^i = \frac{4R}{D}$ and we used the values of ε' and T. Finally, we compute the complexity of this algorithm. We have T iterations taking $\widetilde{O}(\sqrt{\frac{L}{\mu}})$ gradient oracle queries each. Using the value of T and D, we obtain that in total, we call the gradient oracle $\widetilde{O}(\frac{R}{D}\sqrt{\frac{L}{\mu}}) = \widetilde{O}(R^2|\kappa_{\min}|\sqrt{\frac{L}{\mu}}) = \widetilde{O}(\zeta^2\sqrt{\frac{L}{\mu}})$ times, in both of the suggested initializations for D, cf. Algorithm 2.

We conclude by studying the case in which f is not strongly convex. Assume there is a global optimizer x^* and as before let $R \geq d(x_0, x^*)$. Given $\varepsilon > 0$, we use the regularizer $r(x) = \frac{\varepsilon}{2R^2}\Phi_{x_0}(x) = \frac{\varepsilon}{2R^2}d(x_0,x)^2$. Let x_ε^* be the minimizer of $f + r$. By (Martínez-Rubio, 2020, Lemma

21), we have $d(x_0, x_\varepsilon^*) \leq d(x_0, x^*) \leq R$. We run Algorithm 2 on $f + r$, which satisfies that the iterates of the algorithm and the subroutine go no farther than $2R + D/2 < 3R$ from x_ε^*. Indeed, the centers of the balls $\mathcal{X}_k$ are at a distance at most $2R$ from x_ε^* by Proposition 11 and each ball has radius $D/2$. Recall that we are still optimizing over a Hadamard manifold. So in $\bar{B}(x_\varepsilon^*, 3R)$, we have that $f + r$ is strongly g-convex with constant $\frac{\varepsilon}{R^2}$. Moreover, its smoothness constant is $\zeta_{3R} \cdot \frac{\varepsilon}{R^2} + L = O(\zeta \cdot \frac{\varepsilon}{R^2} + L)$, by Fact 2. Hence, the algorithm finds an $\varepsilon/2$ minimizer $x_{T'}$ of $f + r$ after $T' = \widetilde{O}(\zeta^2\sqrt{\zeta + \frac{LR^2}{\varepsilon}})$ queries to the gradient oracle. By definition, it is $d(x_0, x^*) \leq R$ so $r(x^*) \leq \frac{\varepsilon}{2R^2} \cdot R^2 = \frac{\varepsilon}{2}$ and thus $x_{T'}$ is an ε-minimizer of f:

$$f(x_{T'}) \leq f(x_{T'}) + r(x_{T'}) \leq f(x^*) + r(x^*) + \frac{\varepsilon}{2} \leq f(x^*) + \varepsilon.$$

Finally, we note that the condition of the theorem requiring optimizing in $\bar{B}(x^*, 3R)$ can be relaxed to work in $\bar{B}(x^*, c_1 R + c_2)$ for $c_1 > 1$ and $c_2 \in (0, 1/(70R|\kappa_{\min}|))$, by solving the ball subproblems more accurately and in smaller balls. ■

B.1. Details of the subroutine chosen by Algorithm 2 for Line 8 of Algorithm 1

Given a constant F such that $\|\nabla\mathfrak{R}\| \leq F$, in their Proposition 6.1, Criscitiello and Boumal (2021) argue that given an L'-smooth and μ'-strongly g-convex function in a ball of radius $r \leq \min\{\frac{\sqrt{\mu'}}{4\sqrt{L'}|\kappa_{\min}|}, \frac{|\kappa_{\min}|}{4F}\}$[3], the retraction of the function in the ball to the Euclidean space $\hat{h}(\cdot) \stackrel{\text{def}}{=} h \circ \operatorname{Exp}_{x_k}(\cdot)$ is strongly convex and smooth with condition number $\frac{3L'}{\mu'}$. Here, $\frac{1}{F}$ is interpreted as $+\infty$. They assume that the global minimizer is in this ball, but this fact is only used in order to use L'-smoothness to bound the Lipschitz constant of the function by $2rL'$. In our case, the global optimizer is at a distance at most $3R$ from any point in any of our balls $\mathcal{X}_k$, as argued in the previous section. However, we can bound the Lipschitz constant by other means. The functions we will apply this subroutine to have the form $h(y) \stackrel{\text{def}}{=} f(y) + \frac{1}{\lambda}d(x, y)^2$, where x is a point such that $d(x, y) \leq 2D$ for all $y \in \mathcal{X}_k$, cf. Line 8 in Algorithm 1 and (1). Here $D = 2r$ is the diameter of $\mathcal{X}_k$. Using the value of λ, we have that the smoothness of $g : y \mapsto \frac{1}{\lambda}d(x, y)^2$ in the ball $\mathcal{X}_k$ is L and the global minimizer of this function is at most a distance $2D = 4r$. So we can estimate the Lipschitz constant of such an h as

$$\max_{y \in \mathcal{X}_k} \|\nabla h(y)\| \leq \max_{y \in \mathcal{X}_k} \|\nabla f(y)\| + \max_{y \in \mathcal{X}_k} \|\nabla g(y)\| \leq 6RL + 8rL \leq 14RL,$$

where the last inequality uses $r \leq R$ which holds by construction of Algorithm 2. Now, it is enough to satisfy the following inequality in Proposition 6.1 in (Criscitiello and Boumal, 2021) in order to have that the Euclidean pulled-back function has condition number of the same order as h, which is $O(\bar{\zeta})$ for $\mathcal{X}_k$:

$$\frac{7}{9}L'|\kappa_{\min}|r^2 + \frac{3}{2}|\kappa_{\min}|r \max_{y \in \mathcal{X}_k} \|\nabla h(y)\| \leq \frac{\mu'}{2} = \frac{\mu + L/\zeta_{2D}}{2}.$$

Since $r \leq R$, $\zeta_{2D} \leq 2\zeta_D$, $L' = 2L$ and $\mu \geq 0$, it is enough to have $23LrR|\kappa_{\min}| \leq L/(4\zeta_D)$. Note that in Algorithm 2, we ensure $r \leq (92R|\kappa_{\min}|\zeta_{2r})^{-1}$ which satisfies the previous inequality and also the initial requirement in Proposition 6.1 in (Criscitiello and Boumal, 2021).

3. This bound corresponds to the case of Hadamard manifolds. Their statement applies more generally to manifolds of bounded sectional curvature, in which case $|\kappa_{\min}|$ would be substituted by $\max\{|\kappa_{\min}|, \kappa_{\max}\}$.

After this result, we can use Euclidean machinery on $\hat{h} : \mathrm{Log}_{x_{k-1}}(\mathcal{X}_k) \to \mathbb{R}$, namely AGD (Nesterov, 2005) with a warm start in order to satisfy the condition in Line 8 of Algorithm 1. The algorithm requires projecting onto the feasible set, and we note that in our case it is a Euclidean ball so the operation is very simple. Indeed, let $\hat{\mathcal{X}}_k \stackrel{\mathrm{def}}{=} \mathrm{Log}_{x_{k-1}}(\mathcal{X}_k)$ and let $\hat{x} \stackrel{\mathrm{def}}{=} \mathrm{Log}_{x_{k-1}}(x)$, where x is the center of the prox defining g above. By (Lin et al., 2017, Proposition 15) we have that $\hat{x}' \stackrel{\mathrm{def}}{=} \Pi_{\hat{\mathcal{X}}_k}(\Pi_{\hat{\mathcal{X}}_k}(\hat{x}) - \frac{1}{L'}\nabla\hat{h}(\Pi_{\hat{\mathcal{X}}_k}(\hat{x})))$ is a point that satisfies

$$\hat{h}(\hat{x}') - \hat{h}(\hat{y}_h^*) \leq \frac{L'}{2}\|\hat{y}_h^* - \Pi_{\hat{\mathcal{X}}_k}(\hat{x})\|^2 \leq \frac{L'}{2}\|\hat{y}_h^* - \hat{x}\|^2, \tag{12}$$

where $\hat{y}_h^* \stackrel{\mathrm{def}}{=} \arg\min_{\hat{y}\in\hat{\mathcal{X}}_k} \hat{h}(y)$ is the minimizer of $\hat{h}$, that is, the exact prox. By (Nesterov, 2005), the convergence rate of AGD with $\hat{x}'$ as initial point is $O(\sqrt{\frac{L'}{\mu'}}\log(\frac{\hat{h}(\hat{x}')-h(\hat{y}_h^*)}{\hat{\varepsilon}}))$, where the accuracy we will require is $\hat{\varepsilon} \stackrel{\mathrm{def}}{=} \Delta_{k'}\|\hat{y}_h^* - \hat{x}\|^2_{x_{k-1}}/(78\lambda)$, which is less than the $\Delta_{k'}d(\hat{x}, \hat{y}_h^*)^2/(78\lambda)$ accuracy required by Algorithm 1. Here k' is the internal counter for Algorithm 1 and we used the reasoning above yielding that the condition number of $\hat{h}$ is $O(\frac{L'}{\mu'}) = O(\bar{\zeta})$. Using (12), we conclude that it is enough to run AGD for $O(\bar{\zeta}^{\frac{1}{2}}\log(\frac{78\lambda L'}{2\Delta_{k'}})) = \widetilde{O}(\bar{\zeta}^{\frac{1}{2}})$ gradient oracle queries.

Remark 12 *We can make Algorithm 2 work under a weaker assumption than Assumption 5 after a minor modification on the algorithm. Because the algorithm in (Criscitiello and Boumal, 2021) can work with bounded* $\|\nabla\mathfrak{R}\| \leq F$ *for a constant* F*, we can use it as a subroutine in this more generic case. In such a case, the diameter of the balls* $\mathcal{X}_k$ *must be* $D \leq \frac{|\kappa_{\min}|}{2F}$*, and it is enough to change the condition in Line 2 to* $2R \leq \min\{(46R|\kappa_{\min}|\zeta_{2R})^{-1}, |\kappa_{\min}|/(2F)\}$ *and if this condition is not satisfied, then after computing* D *in Line 4, we further update* $D \leftarrow \min\{D, |\kappa_{\min}|/(2F)\}$*. In this way, the condition is satisfied and the geometric penalty is* $\widetilde{O}(\frac{R}{D}) = \widetilde{O}(\zeta^2 + \frac{RF}{|\kappa_{\min}|})$ *instead of* $\widetilde{O}(\zeta^2)$*.*

Appendix C. Geometric lemmas

In this section, we state and prove Lemma 17, which is used in the proof of Theorem 4 to show that the lower bound given by $f(y_k^*) + \langle\tilde{v}_k^y, x - y_k^*\rangle$ that is affine if pulled back to $T_{y_k^*}$ can be bounded by another function, that is affine if pulled back to T_{x_k}. We also include and prove, with some generalizations, some known Riemannian inequalities that are used in Riemannian optimization methods and that we also use. The second part of the following lemma appeared in (Kim and Yang, 2022). Similarly with the second part of the corollary that follows.

In this section, unless otherwise specified, $\mathcal{M}$ is an n-dimensional Riemannian manifold of bounded sectional curvature.

Lemma 13 *Let* $x, y, p \in \mathcal{M}$ *be the vertices of a uniquely geodesic triangle* $\mathcal{T}$ *of diameter* D*, and let* $z^x \in T_x\mathcal{M}$, $z^y \stackrel{\mathrm{def}}{=} \Gamma_x^y(z^x) + \mathrm{Log}_y(x)$*, such that* $y = \mathrm{Exp}_x(rz^x)$ *for some* $r \in [0, 1)$*. If we take vectors* $a^y \in T_y\mathcal{M}$, $a^x \stackrel{\mathrm{def}}{=} \Gamma_y^x(a^y) \in T_x\mathcal{M}$*, then we have the following, for all* $\xi \geq \zeta_D$*:*

$$\begin{aligned}&\|z^y + a^y - \mathrm{Log}_y(p)\|_y^2 + (\delta_D - 1)\|z^y + a^y\|_y^2 \\ &\quad\geq \|z^x + a^x - \mathrm{Log}_x(p)\|_x^2 + (\delta_D - 1)\|z^x + a^x\|_x^2 - \frac{\xi - \delta_D}{2}\left(\frac{r}{1-r}\right)\|a^x\|_x^2,\end{aligned}$$

and

$$\begin{aligned}&\|z^y + a^y - \mathrm{Log}_y(p)\|_y^2 + (\xi - 1)\|z^y + a^y\|_y^2 \\ &\quad \leq \|z^x + a^x - \mathrm{Log}_x(p)\|_x^2 + (\xi - 1)\|z^x + a^x\|_x^2 + \frac{\xi - \delta_D}{2}\left(\frac{r}{1-r}\right)\|a^x\|_x^2.\end{aligned}$$

Proof Let γ be the unique geodesic in $\mathcal{T}$ such that $\gamma(0) = x$ and $\gamma(r) = y$. We have $\gamma'(0) = z^x$. Along γ, we define the vector field $V(t) = \Gamma_0^t(\gamma)(z^x - t\gamma'(0))$. Then, it is $V'(t) = -\gamma'(t)$, and $\|V(t)\| = \|a + (1-t)z^x\|$. We will make use of the potential $w : [0, r] \to \mathbb{R}$ defined as $w(t) = \|\mathrm{Log}_{\gamma(t)}(x) - V(t)\|^2$. We can compute

$$\begin{aligned}\frac{d}{dt}w(t) &= 2\langle D_t(\mathrm{Log}_{\gamma(t)}(x) - V(t)), \mathrm{Log}_{\gamma(t)}(x) - V(t)\rangle \\ &= 2\langle D_t\mathrm{Log}_{\gamma(t)}(x), \mathrm{Log}_{\gamma(t)}(x)\rangle - 2\langle D_t\mathrm{Log}_{\gamma(t)}(x), V(t)\rangle \\ &\quad - 2\langle D_tV(t), \mathrm{Log}_{\gamma(t)}(x)\rangle + 2\langle D_tV(t), V(t)\rangle \\ &= -2\langle D_t(\mathrm{Log}_{\gamma(t)}(x), V(t)\rangle + 2\langle D_tV(t), V(t)\rangle.\end{aligned} \tag{13}$$

Now, we bound the first summand. We use that for the function $\Phi_p(x) = \frac{1}{2}d(x,p)^2$ it holds, for every $\xi \geq \zeta_D$:

$$-\frac{\xi - \delta_D}{2}\|v\|^2 \leq \langle \mathrm{Hess}\,\Phi_p(\gamma(t))[v] - \frac{\xi + \delta_D}{2}v, v\rangle \leq \frac{\xi - \delta_D}{2}\|v\|^2,$$

due to Fact 2. So for $\beta \in \{-1, 1\}$ we obtain the following bound:

$$\begin{aligned}&-2\beta\langle D_t\mathrm{Log}_{\gamma(t)}(x), V(t)\rangle = 2\beta\langle \mathrm{Hess}\,\Phi_p(\gamma(t))[\gamma'(t)], V(t)\rangle \\ &\quad = 2\beta\langle(\ \mathrm{Hess}\,\Phi_p(\gamma(t)) - \frac{\xi + \delta_D}{2}I\ )[\gamma'(t)], V(t)\rangle + \beta\langle(\xi + \delta_D)\gamma'(t), V(t)\rangle \\ &\quad \leq 2\|\ \mathrm{Hess}\,\Phi_p(\gamma(t)) - \frac{\xi + \delta_D}{2}I\| \cdot \|\gamma'(t)\| \cdot \|V(t)\| + \beta\langle(\xi + \delta_D)\gamma'(t), V(t)\rangle \\ &\quad \leq 2\frac{\xi - \delta_D}{2}\|\gamma'(t)\| \cdot \|V(t)\| + \beta\langle(\xi + \delta_D)\gamma'(t), V(t)\rangle \\ &\quad \overset{①}{=} 2\frac{\xi - \delta_D}{2}\|z^x\| \cdot \|a + (1-t)z^x\| + \beta(\xi + \delta_D)\langle z^x, a + (1-t)z^x\rangle\end{aligned}$$

Gauss lemma is used in the last summand of ①. Now, if $\beta = -1$, we have

$$\begin{aligned}&-2\langle D_t\mathrm{Log}_{\gamma(t)}(x), V(t)\rangle \geq -2\frac{\xi - \delta_D}{2}\|z^x\| \cdot \|a + (1-t)z^x\| + (\xi + \delta_D)\langle z^x, a + (1-t)z^x\rangle \\ &\overset{①}{\geq} -\frac{\xi - \delta_D}{2(1-t)}(\|(1-t)z^x\|^2 + \|a + (1-t)z^x\|^2) + (\xi - \delta_D)\langle z^x, a + (1-t)z^x\rangle - 2\delta_D\langle -z^x, a + (1-t)b\rangle \\ &\geq -\frac{\xi - \delta_D}{2(1-t)}(\|a\|^2 + 2\langle a + (1-t)z^x\rangle) + (\xi - \delta_D)\langle z^x, a\rangle - 2\delta_D\langle -z^x, a + (1-t)b\rangle \\ &\geq -\frac{\xi - \delta_D}{2(1-t)}\|a\|^2 - 2\delta_D\langle D_tV(t), V(t)\rangle.\end{aligned} \tag{14}$$

On the other hand, analogously, if $\beta = 1$, we have

$$
\begin{aligned}
-2\langle D_t \mathrm{Log}_{\gamma(t)}(x), V(t)\rangle &\leq 2\frac{\xi-\delta_D}{2}\|z^x\|\cdot\|a+(1-t)z^x\| + (\xi+\delta_D)\langle z^x, a+(1-t)z^x\rangle \\
&\overset{①}{\leq} \frac{\xi-\delta_D}{2(1-t)}(\|(1-t)z^x\|^2 + \|a+(1-t)z^x\|^2) - (\xi-\delta_D)\langle z^x, a+(1-t)z^x\rangle - 2\xi\langle -z^x, a+(1-t)b\rangle \\
&\leq \frac{\xi-\delta_D}{2(1-t)}(\|a\|^2 + 2\langle a+(1-t)z^x\rangle) - (\xi-\delta_D)\langle z^x, a\rangle - 2\xi\langle -z^x, a+(1-t)b\rangle \\
&\leq \frac{\xi-\delta_D}{2(1-t)}\|a\|^2 - 2\xi\langle D_t V(t), V(t)\rangle,
\end{aligned} \tag{15}
$$

where ① is Young's inequality $2cd \leq c^2 + d^2$. Combining (13), (14), (15), we obtain

$$
-\frac{\xi-\delta_D}{2(1-t)}\|a\|^2 - 2(\delta_D-1)\langle D_t V(t), V(t)\rangle \leq \frac{d}{dt}w(t) \leq \frac{\xi-\delta_D}{2(1-t)}\|a\|^2 - 2(\xi-1)\langle D_t V(t), V(t)\rangle.
$$

Integrating between 0 and $r < 1$, it results in

$$
\begin{aligned}
\frac{\xi-\delta_D}{2}\log(1-r)\|a\|^2 - (\delta_D-1)(\|V(r)\|^2 - \|V(0)\|^2) &\leq w(r) - w(0) \\
&\leq -\frac{\xi-\delta_D}{2}\log(1-r)\|a\|^2 - (\xi-1)(\|V(r)\|^2 - \|V(0)\|^2).
\end{aligned}
$$

Using the bound $-\log(1-r) \leq \frac{r}{1-r}$ for $r \in [0,1)$ and using the values of $w(\cdot)$ and $V(\cdot)$, we obtain the result. ■

Corollary 14 *Let $x, y, p \in \mathcal{M}$ be the vertices of a uniquely geodesic triangle of diameter D, and let $z^x \in T_x\mathcal{M}$, $z^y \overset{\mathrm{def}}{=} \Gamma_x^y(z^x) + \mathrm{Log}_y(x)$, such that $y = \mathrm{Exp}_x(rz^x)$ for some $r \in [0,1]$. Then, the following holds*

$$
\|z^y - \mathrm{Log}_y(p)\|^2 + (\delta_D - 1)\|z^y\|^2 \geq \|z^x - \mathrm{Log}_x(p)\|^2 + (\delta_D - 1)\|z^x\|^2,
$$

and

$$
\|z^y - \mathrm{Log}_y(p)\|^2 + (\zeta_D - 1)\|z^y\|^2 \leq \|z^x - \mathrm{Log}_x(p)\|^2 + (\zeta_D - 1)\|z^x\|^2.
$$

Proof Use Lemma 13 with $a^y = 0$. Note that this corollary allows $r = 1$ as well. We obtain this result, by continuity, by taking a limit when $r \to 1$. ■

The following is a lemma that is already known and is used extensively in Riemannian first-order optimization. It turns out it is a special case of Corollary 14.

Corollary 15 (Cosine-Law Inequalities) *For the vertices $x, y, p \in \mathcal{M}$ of a uniquely geodesic triangle of diameter D, we have*

$$
\langle \mathrm{Log}_x(y), \mathrm{Log}_x(p)\rangle \geq \frac{\delta_D}{2}d(x,y)^2 + \frac{1}{2}d(p,x)^2 - \frac{1}{2}d(p,y)^2.
$$

and

$$
\langle \mathrm{Log}_x(y), \mathrm{Log}_x(p)\rangle \leq \frac{\zeta_D}{2}d(x,y)^2 + \frac{1}{2}d(p,x)^2 - \frac{1}{2}d(p,y)^2
$$

Proof This is Corollary 14 for $r = 1$. Indeed, given $y \in \mathcal{T}$ we can use Corollary 14 with $z^x = \mathrm{Log}_x(y)$. Note that in such a case we have $\|z^x\| = d(x,y)$ and $z^y = 0$. Using $\|\mathrm{Log}_y(p)\| = d(y,p)$ and

$$\begin{aligned}\|z^x - \mathrm{Log}_x(p)\| &= \|z^x\|^2 - \langle z^x, \mathrm{Log}_x(p)\rangle + \|\mathrm{Log}_x(p)\|^2 \\ &= d(x,y)^2 - 2\langle \mathrm{Log}_x(y), \mathrm{Log}_x(p)\rangle + d(p,x)^2,\end{aligned}$$

we obtain the result. ■

Remark 16 *Actually if we substitute the constant ζ_D in the previous Corollary 15 by the tighter constant $\zeta_{d(p,x)}$, the result also holds. See (Zhang and Sra, 2016, Lemma 1).*

We now proceed to prove a lemma that intuitively says that solving the exact proximal point problem can be used to lower bound f. Compare the result of the following lemma with the Euclidean equality $\langle g, p-y\rangle = \langle g, p-x\rangle + \|g\|^2$, for $g = x - y$ and $x, y, p \in \mathbb{R}^n$.

Lemma 17 *Let $x, y, p \in \mathcal{M}$ be the vertices of a uniquely geodesic triangle of diameter D. Define the vectors $g \stackrel{\text{def}}{=} \mathrm{Log}_y(x) \in T_y\mathcal{M}$ and $g^x = \Gamma_y^x(g) = -\mathrm{Log}_x(y) \in T_x\mathcal{M}$. Then we have*

$$\langle g, \mathrm{Log}_y(p)\rangle \geq \langle g^x, \mathrm{Log}_x(p)\rangle + \delta_D\|g\|^2,$$

and

$$\langle g, \mathrm{Log}_y(p)\rangle \leq \langle g^x, \mathrm{Log}_x(p)\rangle + \zeta_D\|g\|^2.$$

Proof Using the definition of g, we have ① below, by the first part of Corollary 15:

$$\begin{aligned}\langle g, \mathrm{Log}_y(p)\rangle &\overset{①}{\geq} \frac{\delta_D}{2}\|g\|^2 + \frac{d(y,p)^2}{2} - \frac{d(x,p)^2}{2} \\ &\overset{②}{\geq} \langle g^x, \mathrm{Log}_x(p)\rangle + \delta_D\|g^x\|^2,\end{aligned}$$

and in ② we used Corollary 15 again but with a different choice of vertices so we have $\frac{d(y,p)^2}{2} \geq \frac{\delta_D}{2}\|g^x\|^2 + \frac{d(x,p)^2}{2} + \langle g^x, \mathrm{Log}_x(p)\rangle$.

The proof of the second part is analogous: using the definition of g, we have ① below, by the second part of Corollary 15:

$$\begin{aligned}\langle g, \mathrm{Log}_y(p)\rangle &\overset{①}{\leq} \frac{\zeta_D}{2}\|g\|^2 + \frac{d(y,p)^2}{2} - \frac{d(x,p)^2}{2} \\ &\overset{②}{\leq} \langle g^x, \mathrm{Log}_x(p)\rangle + \zeta_D\|g^x\|^2,\end{aligned}$$

and in ② we used Corollary 15 again but with a different choice of vertices so we have $\frac{d(y,p)^2}{2} \leq \frac{\zeta_D}{2}\|g^x\|^2 + \frac{d(x,p)^2}{2} + \langle g^x, \mathrm{Log}_x(p)\rangle$. ■